IF : 4.547 | IC Value 80.26

VOLUME-6, ISSUE-7, JULY-2017 • ISSN No 2277 - 8160

Original Research Paper

Engineering

DUAL ETHERNET FRAME DETECTOR LAYOUT & VERIFICATION

Sharat Chandra Musham — Qualcomm Austin

Abhishek Banerjee — Tata Consultancy Services

Bala Chintamneedi — University of Texas Austin

KEYWORDS :

Introduction

The purpose of the project is to place the cells and route in order to implement the Ethernet Frame detector developed for project 3. To achieve the over 100 cell requirement, we implemented Dual Ethernet Frame Detector (DEFD). The idea is to design and implement a circuit which has practical significance and yet simple to design and implement so that the focus will be more on learning the tools. Finally Pathmill was used to find out the worst delays from inputs to outputs.

1. Functionality of Ethernet Frame Detector

The Ethernet Frame detector is often found in Ethernet Switches and it not common to find a 24 port Ethernet switch. We developed a Dual Ethernet Frame Detector, which detects the starting of Ethernet frames from two different Ethernet ports, Fig 1 shows the symbolic view of the cell. Each detector has one input and one output apart from the clock and reset inputs to conquer the FSM modeling.

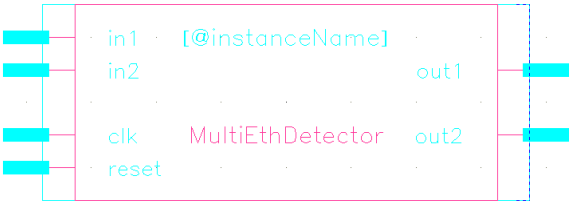


Fig. 1

The detector applies FSM to the input bits on the frame and detects the patthern AA AA AA AA AA AA AA AB, which is the Pre-amble of the Ethernet frame. The ouput is raised to high when the input follows the above mentioned sequence and the output is zero in all other cases. The sequence detector is modeled as FSM because it needs to remember the state of the machine to influence the output at any given time. The output is used for the rest of the circuit as a trigger. There should be additional logic to receive the packet calculate the CRC and verify the CRC as part of the packet. Out interest is to focus on the detector.

2. Input vs Outputs

There are two inputs in1 and in2 apart from clk and reset as mentioned in Fig 1. out1 represents the output for in1 and out2 represents the output for in2. The Ethernet frames are fed to in1 and in2 in parallel. The following is the schematics of the logic circuit.

3. Trade-offs

The circuit we implemented is not the best possible logic circuit or the circuit with minimal logic. Synopsys was used to synthesize the cells and was relied upon. No effort was made to apply any more optimizations. We also didn't pay much attention to power consumption. The performance of individual cells designed as part of project 5 has a direct effect on the DEFD cell. We minimized the D flip-flop to height of 4.75 microns and applied the same height to all the cells. The following is the report from Synopsys about the cell count and power consumption.

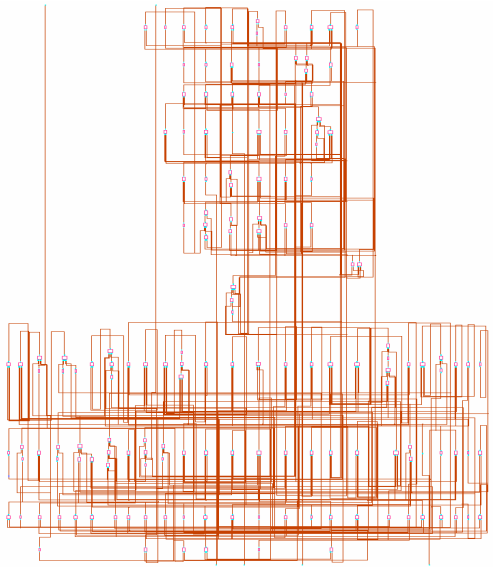


Fig. 2

4.1 Cell count report from Synopsys

```
****************************************
Report : cell
Design : MultiEthDetector
Version: V-2003.12
Date   : Fri Nov 25 03:40:13 2005
****************************************

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

Cell	Reference	Library	Area	Attributes
U15	inv	library	1.000000	
U16	nor3	library	1.000000	
U17	nand3	library	1.000000	
U18	nand2	library	1.000000	
U19	nand3	library	1.000000	

U20	inv	library	1.000000
U21	oai12	library	2.000000
U22	nand4	library	1.000000
U23	oai12	library	2.000000
U24	nor2	library	1.000000
U25	inv	library	1.000000
U26	aoi22	library	2.000000
U27	oai12	library	2.000000
U28	nand2	library	1.000000
U29	nand2	library	1.000000
U30	inv	library	1.000000
U31	nand4	library	1.000000
U32	nand4	library	1.000000
U33	aoi22	library	2.000000
U34	oai22	library	2.000000
U35	nand4	library	1.000000
U36	nand2	library	1.000000
U37	nand3	library	1.000000
U38	nand2	library	1.000000
U39	inv	library	1.000000
U40	nand2	library	1.000000
U41	nand3	library	1.000000
U42	nand2	library	1.000000
U43	inv	library	1.000000
U44	nand2	library	1.000000
U45	inv	library	1.000000
U46	inv	library	1.000000
U47	nand2	library	1.000000
U48	oai12	library	2.000000
U49	nor3	library	1.000000
U50	aoi22	library	2.000000
U51	nand3	library	1.000000
U52	aoi22	library	2.000000
U53	nor2	library	1.000000
U54	inv	library	1.000000
U55	nor2	library	1.000000
U56	nor2	library	1.000000
U57	nand3	library	1.000000
U58	nor2	library	1.000000
U59	inv	library	1.000000
U60	nand2	library	1.000000
U61	inv	library	1.000000
U62	nor2	library	1.000000
U63	nand2	library	1.000000
U64	inv	library	1.000000
U65	inv	library	1.000000
U66	oai12	library	2.000000
U67	nand3	library	1.000000
U68	inv	library	1.000000
U69	nand2	library	1.000000
U70	inv	library	1.000000
U71	oai12	library	2.000000
U72	nand4	library	1.000000
U73	nand4	library	1.000000
U74	inv	library	1.000000
U75	inv	library	1.000000
U76	nand4	library	1.000000
U77	inv	library	1.000000
U78	nand4	library	1.000000
U79	inv	library	1.000000
U80	oai12	library	2.000000
U81	aoi22	library	2.000000
U82	nand3	library	1.000000
U83	nand2	library	1.000000
U84	aoi22	library	2.000000
U85	nand3	library	1.000000
U86	inv	library	1.000000
U87	nand4	library	1.000000
U88	nor3	library	1.000000
U89	nand2	library	1.000000
U90	nand2	library	1.000000
U91	inv	library	1.000000
U92	aoi22	library	2.000000
U93	nor2	library	1.000000
U94	nand4	library	1.000000
U95	nor3	library	1.000000
U96	oai22	library	2.000000
U97	aoi12	library	2.000000
U98	inv	library	1.000000
U99	nand2	library	1.000000
U100	aoi12	library	2.000000
U101	nor2	library	1.000000
U102	inv	library	1.000000
U103	inv	library	1.000000
U104	inv	library	1.000000
U105	aoi22	library	2.000000
U106	nor2	library	1.000000
U107	oai12	library	2.000000
U108	nand2	library	1.000000
U109	oai22	library	2.000000
U110	aoi12	library	2.000000
U111	nor2	library	1.000000
U112	aoi12	library	2.000000
U113	nor2	library	1.000000
U114	nand2	library	1.000000
U115	aoi22	library	2.000000
U116	nor3	library	1.000000
U117	nor2	library	1.000000
U118	nor2	library	1.000000
U119	nand2	library	1.000000
U120	nor2	library	1.000000
U121	nand3	library	1.000000
U122	aoi22	library	2.000000
U123	nor2	library	1.000000
U124	nor2	library	1.000000
U125	nand2	library	1.000000
U126	nand3	library	1.000000
U127	inv	library	1.000000
U128	nand4	library	1.000000
U129	nand3	library	1.000000
U130	inv	library	1.000000
U131	nand2	library	1.000000
U132	nand2	library	1.000000
U133	inv	library	1.000000
U134	nand2	library	1.000000
U135	inv	library	1.000000
U136	nand2	library	1.000000
U137	oai12	library	2.000000
U138	nand3	library	1.000000
U139	nor2	library	1.000000
U140	nand2	library	1.000000
U141	nor2	library	1.000000
U142	inv	library	1.000000
U143	nand2	library	1.000000
U144	inv	library	1.000000
U145	nand3	library	1.000000
U146	nor2	library	1.000000
U147	inv	library	1.000000
U148	inv	library	1.000000
U149	inv	library	1.000000
U150	nand2	library	1.000000
U151	oai12	library	2.000000
U152	nand3	library	1.000000
U153	inv	library	1.000000
U154	nand2	library	1.000000
U155	inv	library	1.000000
__tmp51/state_reg<0>	dff	library	7.000000 n
__tmp51/state_reg<1>	dff	library	7.000000 n

```
__tmp51/state_reg<2>    dff        library      7.000000 n
__tmp51/state_reg<3>    dff        library      7.000000 n
__tmp51/state_reg<4>    dff        library      7.000000 n
__tmp51/state_reg<5>    dff        library      7.000000 n
__tmp51/z_reg           dff        library      7.000000 n
__tmp52/state_reg<0>    dff        library      7.000000 n
__tmp52/state_reg<1>    dff        library      7.000000 n
__tmp52/state_reg<2>    dff        library      7.000000 n
__tmp52/state_reg<3>    dff        library      7.000000 n
__tmp52/state_reg<4>    dff        library      7.000000 n
__tmp52/state_reg<5>    dff        library      7.000000 n
__tmp52/z_reg           dff        library      7.000000 n

--------------------------------------------------------------------------------
Total 155 cells                                  266.000000
```

4.1 Area report from Synopsys

```
****************************************
Report : area
Design : MultiEthDetector
Version: V-2003.12
Date   : Fri Nov 25 03:41:28 2005
****************************************

Library(s) Used:
library (File: /home/002/s/sc/scm042000/cad/synopsys/library.db)

Number of ports:          6
Number of nets:           159
Number of cells:          155
Number of references:     11

Combinational area:      168.000000
Noncombinational area:    98.000000
Net Interconnect area:    undefined  (No wire load specified)

Total cell area:         266.000000
Total area:              undefined
```

4.2 Power report from Synopsys

```
****************************************
Report : power
        -analysis_effort low
Design : MultiEthDetector
Version: V-2003.12
Date   : Fri Nov 25 03:42:07 2005
****************************************


Library(s) Used:
library (File:
 /home/002/s/sc/scm042000/cad/synopsys/library.db)

Warning: The library cells used by your design are not characterized
for internal power. (PWR-26)

Operating Conditions:
Wire Load Model Mode: top

Global Operating Voltage = 5
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 1.000000pf
 Time Units = 1ns
Dynamic Power Units = 1mW    (derived from V,C,T units)
Leakage Power Units = Unitless
Cell Internal Power  =  0.0000 mW   (0%)
Net Switching Power  = 598.1239 mW  (100%)
                       ---------
Total Dynamic Power    = 598.1239 mW  (100%)
Cell Leakage Power     =  0.0000
```

4. Simulation on Model-sim

The synthesized verilog file obtained from Synopsys is tested using ModelSim. The following are the waveforms indicating inputs, outputs and the FSM states. The verilog code is added in Appendix B.

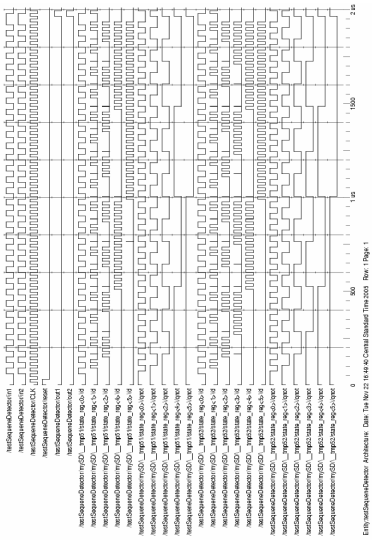


Fig. Model-Sim output

5. Routing using Encounter

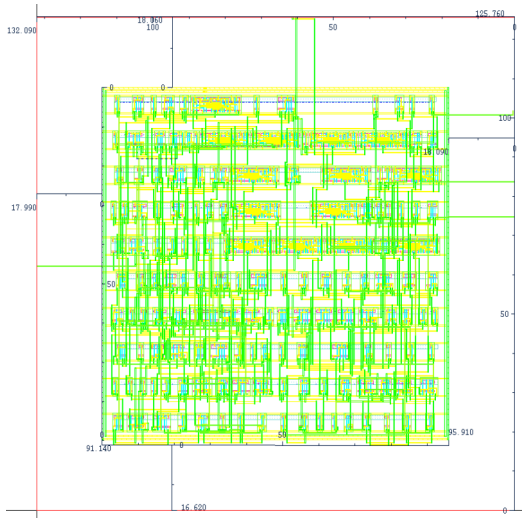
Using Encounter was straight forward. We had to develop .def and .asc files from Cadence and feed these files to Encounter along with synthesized verilog file. We used the tool to add ring and add strips and finally obtained a .def file which consists of the routing information.

We imported the .def file to Cadence, which caused a lot of DRC errors.

6. DRC and LVS reports

There were may DRC errors in the DEFD cell generated by Encounter. The significant errors are NW spacing and M3 off grid errors. These errors were cleared patiently and ran LVS. The LVS matched schematic with the extracted view perfectly. The reports from Cadence indicating DRC completion and LVS match are given below

7.1 DEFD cell view



7.2 DRC report

\o executing: rx_strad_di = geomStraddle(rx di)
\o rx_in_di = geomInside(rx di)
\o executing: err381b = geomAndNot(geomSize(di -0.34) rx_di)
\o executing: drc(di rx (enc < 0.1) "GR381: RX within DI >= 0.100 um.")
\o executing: saveDerived(rx_strad_di "GR381: RX straddling DI found!")
\o executing: saveDerived(err381b "GR381b: RX within DI <= 0.340 um.")
\o executing: nwc_encl = geomAndNot(geomHoles(nwcont) nwcont)
\o executing: die_x031 = geomSize(geomGetEdge(di over bkgnd) 0.31)
\o executing: err382 = geomOutside(die_x031 nwcont)
\o executing: err382rng = geomAndNot(di nwc_encl)
\o executing: saveDerived(err382 "GR382b: RX(NWcont) to DI space <= 0.300 um.")
\o executing: saveDerived(err382rng "GR382b: DI not within an RX(NWcont) hole found!")
\o executing: err383 = geomAndNot(di geomEnclose(di rx))
\o executing: saveDerived(err383 "GR383: DI not enclosing an RX shape found!")
\o executing: err384a = geomButtOrOver(di dg)
\o executing: err384b = geomButtOrOver(di pc)
\o executing: err384c = geomButtOrOver(di op)
\o executing: saveDerived(err384a "GR384: DI touching DG found!.")
\o executing: saveDerived(err384b "GR384: DI touching PC found!.")
\o executing: saveDerived(err384c "GR384: DI touching OP found!.")
\o executing: drc(rx_in_di (width < 1.4) "GR385: RX width (when RX is within DI) >= 1.4 um.")
\o executing: err387 = geomAndNot(rx_in_di nw)
\o executing: nw_in_error = geomAndNot(geomSize(nw_touch_di 1.0) geomButtOrOver(geomSize(nw_touch_...
\o executing: err387b = geomAndNot(nw_in_error nw_touch_di)
\o executing: saveDerived(err387 "GR387: (RX within DI) must be within NW ")
\o executing: saveDerived(err387b "GR387b: This (NW touching DI) to RX substrate contact max spac...
\o executing: nwc_di = geomAnd(nwcont nw_di)
\o executing: drc(nwc_di bp (sep < 0.12) "GR388: (RX(NWcont) over NW(touching DI)) to BP space >= ...
\o executing: nwdi_x100 = geomSize(nw_di 1.0)
\o executing: sxc_over_nwdix = geomAnd(sxcont nwdi_x100)
\o executing: drc(bp sxc_over_nwdix (enc < 0.12) "GR389: (RX(sxcont) over (NW(touch DI) +1.0)) wit...
\o executing: esd_diffusions = geomButtOrOver(rx esdummy)
\o executing: drc(esd_diffusions ca (enc < 0.14) "GRESD14f: CA within (RX touching ESDUMMY) >= 0.1...
\o executing: opesdummy = geomAnd(op geomOr(esd_cdm esdummy))
\o executing: esd26_opx = geomSize(opesdummy 0.25)
\o executing: erresd20 = geomOutside(geomSize(opesdummy 0.25) pc)
\o executing: drc(opesdummy pc (sep < 0.24) "GRESD20: (OP under (ESDUMMY or ESD_CDM)) to PC space ...
\o executing: saveDerived(erresd20 "GRESD20: (OP under (ESDUMMY or ESD_CDM)) to PC space == 0.24 u...
\o executing: erresd21bp = geomButtOrOver(opesdummy bp)
\o executing: erresd21nw = geomButtOrOver(opesdummy nw)
\o executing: saveDerived(erresd21bp "GRESD21: (OP under (ESDUMMY or ESD_CDM)) cannot touch BP.")
\o executing: saveDerived(erresd21nw "GRESD21: (OP under (ESDUMMY or ESD_CDM)) cannot touch NW.")
\o executing: erresd22 = geomOutside(opesdummy rx)
\o executing: erresd22x = geomOverlap(geomAnd(rx opesdummy) geomAnd(geomSize(geomAndNot(rx opesdum...
\o executing: saveDerived(erresd22 "GRESD22: (OP under (ESDUMMY or ESD_CDM)) must touch RX.")
\o executing: saveDerived(erresd22x "GRESD22: (OP under (ESDUMMY or ESD_CDM)) must divide RX into ...
\o executing: esd_drain = geomSize(geomSize(geomButtOrOver(opesdummy rx) -0.22) 0.22)
\o executing: esd_source = geomAndNot(geomButtOrOver(opesdummy rx) esd_drain)
\o executing: drc(esd_source (width < 0.44) "GRESD23: (OP under (ESDUMMY or ESD_CDM)) width (sourc...
\o executing: esd_drain_dg = geomAnd(esd_drain dg)
\o executing: drc(esd_drain (width < 3.0) "GRESD24: (OP under (ESDUMMY or ESD_CDM)) width (drain) ...
\o executing: drc(esd_drain_dg (width < 5.0) "GRESD24: (OP under (ESDUMMY or ESD_CDM)) width (drai...
\o executing: esd_pc = geomAnd(pc geomButtOrOver(rx opesdummy))
\o executing: esd_edges = geomAnd(geomSize(geomGetEdge(geomSize(esd_pc 0.24) butting op) 0.005) rx)
\o executing: erresd25 = geomEnclose(geomSize(esd_pc 0.25) esd_edges (keep < 2))
\o erresd25x = geomEnclose(geomSize(esd_pc 0.25) esd_edges (keep > 2))
\o executing: erresd25d = geomOverlap(geomSize(esd_pc 0.25) esd_drain (keep > 1))
\o executing: erresd25s = geomOverlap(geomSize(esd_pc 0.25) esd_source (keep > 1))
\o executing: saveDerived(erresd25 "GRESD25: (ESD gate x0.24um) must touch two (OP under (ESDUMMY ...
\o executing: saveDerived(erresd25x "GRESD25: (ESD gate x0.24um) can only touch two OP under (ESDU...
\o executing: saveDerived(erresd25d "GRESD25: (ESD gate x0.24um) must touch one drain (OP width > ...
\o executing: saveDerived(erresd25s "GRESD25: (ESD gate x0.24um) must touch one source (OP width =...
\o executing: erresd26 = geomOutside(esd26_opx esd_pc)
\o executing: erresd26x = geomOverlap(esd26_opx esd_pc (keep > 1))
\o executing: saveDerived(erresd26 "GRESD26: ((OP under (ESDUMMY or ESD_CDM)) x0.30um) not touchin...
\o executing: saveDerived(erresd26x "GRESD26: ((OP under (ESDUMMY or ESD_CDM)) x0.30um) can only t...
\o executing: errpn001_ma = geomStraddle(ma logobnd)
\o executing: errpn001_bf = geomStraddle(bf logobnd)
\o executing: errpn001_bfmoat = geomStraddle(bfmoat logobnd)
\o executing: errpn001_bh = geomStraddle(bh logobnd)
\o executing: errpn001_bn = geomStraddle(bn logobnd)
\o executing: errpn001_bp = geomStraddle(bp logobnd)
\o executing: errpn001_ca = geomStraddle(ca logobnd)
\o executing: errpn001_cabar = geomStraddle(cabar logobnd)
\o executing: errpn001_de = geomStraddle(de logobnd)
\o executing: errpn001_df = geomStraddle(df logobnd)
\o executing: errpn001_dg = geomStraddle(dg logobnd)
\o executing: errpn001_dv = geomStraddle(dv logobnd)
\o executing: errpn001_ly = geomStraddle(ly logobnd)
\o executing: errpn001_m1 = geomStraddle(m1 logobnd)
\o executing: errpn001_m1chexl = geomStraddle(m1chexl logobnd)
\o executing: errpn001_m2 = geomStraddle(m2 logobnd)
\o executing: errpn001_m2chexl = geomStraddle(m2chexl logobnd)
\o executing: errpn001_m3 = geomStraddle(m3 logobnd)
\o executing: errpn001_m3chexl = geomStraddle(m3chexl logobnd)
\o executing: errpn001_m4 = geomStraddle(m4 logobnd)
\o executing: errpn001_m4chexl = geomStraddle(m4chexl logobnd)
\o executing: errpn001_m5 = geomStraddle(m5 logobnd)
\o executing: errpn001_m5chexl = geomStraddle(m5chexl logobnd)
\o executing: errpn001_m6 = geomStraddle(m6 logobnd)
\o executing: errpn001_m6chexl = geomStraddle(m6chexl logobnd)

\o executing: errpn001_mq = geomStraddle(mq logobnd)
\o executing: errpn001_mqchexl = geomStraddle(mqchexl logobnd)
\o executing: errpn001_mg = geomStraddle(mg logobnd)
\o executing: errpn001_mgchexl = geomStraddle(mgchexl logobnd)
\o executing: errpn001_nr = geomStraddle(nr logobnd)
\o executing: errpn001_nv = geomStraddle(nv logobnd)
\o executing: errpn001_pv = geomStraddle(pv logobnd)
\o executing: errpn001_nw = geomStraddle(nw logobnd)
\o executing: errpn001_nwasp = geomStraddle(nwasp logobnd)
\o executing: errpn001_op = geomStraddle(op logobnd)
\o executing: errpn001_pc = geomStraddle(pc logobnd)
\o executing: errpn001_pcing = geomStraddle(pcing logobnd)
\o executing: errpn001_pd = geomStraddle(pd logobnd)
\o executing: errpn001_ph = geomStraddle(ph logobnd)
\o executing: errpn001_pi = geomStraddle(pi logobnd)
\o executing: errpn001_rr = geomStraddle(rr logobnd)
\o executing: errpn001_rx = geomStraddle(rx logobnd)
\o executing: errpn001_lv = geomStraddle(lv logobnd)
\o executing: errpn001_var = geomStraddle(var logobnd)
\o executing: errpn001_v1 = geomStraddle(v1 logobnd)
\o executing: errpn001_v1bar = geomStraddle(v1bar logobnd)
\o executing: errpn001_v2 = geomStraddle(v2 logobnd)
\o executing: errpn001_v2bar = geomStraddle(v2bar logobnd)
\o executing: errpn001_v3 = geomStraddle(v3 logobnd)
\o executing: errpn001_v3bar = geomStraddle(v3bar logobnd)
\o executing: errpn001_v4 = geomStraddle(v4 logobnd)
\o executing: errpn001_v4bar = geomStraddle(v4bar logobnd)
\o executing: errpn001_v5 = geomStraddle(v5 logobnd)
\o executing: errpn001_v5bar = geomStraddle(v5bar logobnd)
\o executing: errpn001_vl = geomStraddle(vl logobnd)
\o executing: errpn001_vlbar = geomStraddle(vlbar logobnd)
\o executing: errpn001_vq = geomStraddle(vq logobnd)
\o executing: errpn001_vqbar = geomStraddle(vqbar logobnd)
\o executing: errpn001_vg = geomStraddle(vg logobnd)
\o executing: errpn001_vgbar = geomStraddle(vgbar logobnd)
\o executing: errpn001_f1 = geomStraddle(f1 logobnd)
\o executing: errpn001_f1bar = geomStraddle(f1bar logobnd)
\o executing: errpn001_ft = geomStraddle(ft logobnd)
\o executing: errpn001_ftbar = geomStraddle(ftbar logobnd)
\o executing: errpn001_fy = geomStraddle(fy logobnd)
\o executing: errpn001_fybar = geomStraddle(fybar logobnd)
\o executing: errpn001_zerovt = geomStraddle(zerovt logobnd)
\o executing: saveDerived(errpn001_ma "GRPN001: Shape straddling LOGOBND found! (MA)")
\o executing: saveDerived(errpn001_bf "GRPN001: Shape straddling LOGOBND found! (BF)")
\o executing: saveDerived(errpn001_bfmoat "GRPN001: Shape straddling LOGOBND found! (BFMOAT)")
\o executing: saveDerived(errpn001_zerovt "GRPN001: Shape straddling LOGOBND found! (ZEROVT)")
\o executing: saveDerived(errpn001_bh "GRPN001: Shape straddling LOGOBND found! (BH)")
\o executing: saveDerived(errpn001_bn "GRPN001: Shape straddling LOGOBND found! (BN)")
\o executing: saveDerived(errpn001_bp "GRPN001: Shape straddling LOGOBND found! (BP)")
\o executing: saveDerived(errpn001_ca "GRPN001: Shape straddling LOGOBND found! (CA)")
\o executing: saveDerived(errpn001_cabar "GRPN001: Shape straddling LOGOBND found! (CABAR)")
\o executing: saveDerived(errpn001_de "GRPN001: Shape straddling LOGOBND found! (DE)")
\o executing: saveDerived(errpn001_df "GRPN001: Shape straddling LOGOBND found! (DF)")
\o executing: saveDerived(errpn001_dg "GRPN001: Shape straddling LOGOBND found! (DG)")
\o executing: saveDerived(errpn001_dv "GRPN001: Shape straddling LOGOBND found! (DV)")
\o executing: saveDerived(errpn001_ly "GRPN001: Shape straddling LOGOBND found! (LY)")
\o executing: saveDerived(errpn001_m1 "GRPN001: Shape straddling LOGOBND found! (M1)")
\o executing: saveDerived(errpn001_m1chexl "GRPN001: Shape straddling LOGOBND found! (M1CHEXCL)")
\o executing: saveDerived(errpn001_m2 "GRPN001: Shape straddling LOGOBND found! (M2)")
\o executing: saveDerived(errpn001_m2chexl "GRPN001: Shape straddling LOGOBND found! (M2CHEXCL)")
\o executing: saveDerived(errpn001_m3 "GRPN001: Shape straddling LOGOBND found! (M3)")
\o executing: saveDerived(errpn001_m3chexl "GRPN001: Shape straddling LOGOBND found! (M3CHEXCL)")
\o executing: saveDerived(errpn001_m4 "GRPN001: Shape straddling LOGOBND found! (M4)")
\o executing: saveDerived(errpn001_m4chexl "GRPN001: Shape straddling LOGOBND found! (M4CHEXCL)")
\o executing: saveDerived(errpn001_m5 "GRPN001: Shape straddling LOGOBND found! (M5)")
\o executing: saveDerived(errpn001_m5chexl "GRPN001: Shape straddling LOGOBND found! (M5CHEXCL)")
\o executing: saveDerived(errpn001_m6 "GRPN001: Shape straddling LOGOBND found! (M6)")
\o executing: saveDerived(errpn001_m6chexl "GRPN001: Shape straddling LOGOBND found! (M6CHEXCL)")
\o executing: saveDerived(errpn001_mq "GRPN001: Shape straddling LOGOBND found! (MQ)")
\o executing: saveDerived(errpn001_mqchexl "GRPN001: Shape straddling LOGOBND found! (MQCHEXCL)")
\o executing: saveDerived(errpn001_mg "GRPN001: Shape straddling LOGOBND found! (MG)")
\o executing: saveDerived(errpn001_mgchexl "GRPN001: Shape straddling LOGOBND found! (MGCHEXCL)")
\o executing: saveDerived(errpn001_nr "GRPN001: Shape straddling LOGOBND found! (NR)")
\o executing: saveDerived(errpn001_nv "GRPN001: Shape straddling LOGOBND found! (NV)")
\o executing: saveDerived(errpn001_pv "GRPN001: Shape straddling LOGOBND found! (PV)")
\o executing: saveDerived(errpn001_nw "GRPN001: Shape straddling LOGOBND found! (NW)")
\o executing: saveDerived(errpn001_nwasp "GRPN001: Shape straddling LOGOBND found! (NWASP)")
\o executing: saveDerived(errpn001_op "GRPN001: Shape straddling LOGOBND found! (OP)")
\o executing: saveDerived(errpn001_pc "GRPN001: Shape straddling LOGOBND found! (PC)")
\o executing: saveDerived(errpn001_pcing "GRPN001: Shape straddling LOGOBND found! (PCING)")
\o executing: saveDerived(errpn001_pd "GRPN001: Shape straddling LOGOBND found! (PD)")
\o executing: saveDerived(errpn001_ph "GRPN001: Shape straddling LOGOBND found! (PH)")
\o executing: saveDerived(errpn001_pi "GRPN001: Shape straddling LOGOBND found! (PI)")
\o executing: saveDerived(errpn001_rr "GRPN001: Shape straddling LOGOBND found! (RR)")
\o executing: saveDerived(errpn001_rx "GRPN001: Shape straddling LOGOBND found! (RX)")
\o executing: saveDerived(errpn001_lv "GRPN001: Shape straddling LOGOBND found! (LV)")
\o executing: saveDerived(errpn001_var "GRPN001: Shape straddling LOGOBND found! (VAR)")
\o executing: saveDerived(errpn001_v1 "GRPN001: Shape straddling LOGOBND found! (V1)")
\o executing: saveDerived(errpn001_v1bar "GRPN001: Shape straddling LOGOBND found! (V1BAR)")
\o executing: saveDerived(errpn001_v2 "GRPN001: Shape

straddling LOGOBND found! (V2)")
\o executing: saveDerived(errpn001_v2bar "GRPN001: Shape straddling LOGOBND found! (V2BAR)")
\o executing: saveDerived(errpn001_v3 "GRPN001: Shape straddling LOGOBND found! (V3)")
\o executing: saveDerived(errpn001_v3bar "GRPN001: Shape straddling LOGOBND found! (V3BAR)")
\o executing: saveDerived(errpn001_v4 "GRPN001: Shape straddling LOGOBND found! (V4)")
\o executing: saveDerived(errpn001_v4bar "GRPN001: Shape straddling LOGOBND found! (V4BAR)")
\o executing: saveDerived(errpn001_v5 "GRPN001: Shape straddling LOGOBND found! (V5)")
\o executing: saveDerived(errpn001_v5bar "GRPN001: Shape straddling LOGOBND found! (V5BAR)")
\o executing: saveDerived(errpn001_vl "GRPN001: Shape straddling LOGOBND found! (VL)")
\o executing: saveDerived(errpn001_vlbar "GRPN001: Shape straddling LOGOBND found! (VLBAR)")
\o executing: saveDerived(errpn001_vq "GRPN001: Shape straddling LOGOBND found! (VQW)")
\o executing: saveDerived(errpn001_vqbar "GRPN001: Shape straddling LOGOBND found! (VQBAR)")
\o executing: saveDerived(errpn001_vg "GRPN001: Shape straddling LOGOBND found! (VG)")
\o executing: saveDerived(errpn001_vgbar "GRPN001: Shape straddling LOGOBND found! (VGBAR)")
\o executing: saveDerived(errpn001_f1 "GRPN001: Shape straddling LOGOBND found! (F1)")
\o executing: saveDerived(errpn001_f1bar "GRPN001: Shape straddling LOGOBND found! (F1BAR)")
\o executing: saveDerived(errpn001_ft "GRPN001: Shape straddling LOGOBND found! (FT)")
\o executing: saveDerived(errpn001_ftbar "GRPN001: Shape straddling LOGOBND found! (FTBAR)")
\o executing: saveDerived(errpn001_fy "GRPN001: Shape straddling LOGOBND found! (FY)")
\o executing: saveDerived(errpn001_fybar "GRPN001: Shape straddling LOGOBND found! (FY)")
\o executing: errpn101 = geomInside(logobnd geomSize(chipedge -50.01))
\o executing: saveDerived(errpn101 "GRPN101: LOGOBND must be within CHIPEDGE (maximum) < 50.0 um.")
\o executing: ca_logo = geomAnd(ca_all logobnd)
\o executing: drc(ca_logo (sep < 0.28) "GRPN553: CA to CA space (logo) >= 0.280 um.")
\o executing: m1_logo = geomAnd(m1 logobnd)
\o executing: drc(m1_logo (sep < 0.28) "GRPN502: M1 to M1 space (logo) >= 0.280 um.")
\o drc(m1_logo (notch < 0.28) "GRPN502: M1 to M1 notch (logo) >= 0.280 um.")
\o executing: v1_all = geomOr(v1 v1bar)
\o executing: v2_all = geomOr(v2 v2bar)
\o executing: v3_all = geomOr(v3 v3bar)
\o executing: v4_all = geomOr(v4 v4bar)
\o v5_all = geomOr(v4 v4bar)
\o executing: v1_logo = geomAnd(v1_all logobnd)
\o executing: v2_logo = geomAnd(v2_all logobnd)
\o executing: v3_logo = geomAnd(v3_all logobnd)
\o executing: v4_logo = geomAnd(v4_all logobnd)
\o executing: v5_logo = geomAnd(v5_all logobnd)
\o executing: drc(v1_logo (sep < 0.28) "GRPN553: V1 to V1 space (logo) >= 0.280 um.")
\o executing: drc(v2_logo (sep < 0.28) "GRPN553: V2 to V2 space (logo) >= 0.280 um.")
\o executing: drc(v3_logo (sep < 0.28) "GRPN553: V3 to V3 space (logo) >= 0.280 um.")
\o executing: drc(v4_logo (sep < 0.28) "GRPN553: V4 to V4 space (logo) >= 0.280 um.")
\o executing: drc(v5_logo (sep < 0.28) "GRPN553: V5 to V5 space (logo) >= 0.280 um.")
\o executing: m2_logo = geomAnd(m2 logobnd)
\o executing: m3_logo = geomAnd(m3 logobnd)
\o executing: m4_logo = geomAnd(m4 logobnd)
\o executing: m5_logo = geomAnd(m5 logobnd)
\o executing: m6_logo = geomAnd(m6 logobnd)
\o executing: drc(m2_logo (sep < 0.28) "GRPN602: M2 to M2 space (logo) >= 0.280 um.")
\o drc(m2_logo (notch < 0.28) "GRPN602: M2 to M2 notch (logo) >= 0.280 um.")
\o executing: drc(m3_logo (sep < 0.28) "GRPN602: M3 to M3 space (logo) >= 0.280 um.")
\o drc(m3_logo (notch < 0.28) "GRPN602: M3 to M3 notch (logo) >= 0.280 um.")
\o executing: drc(m4_logo (sep < 0.28) "GRPN602: M4 to M4 space (logo) >= 0.280 um.")
\o drc(m4_logo (notch < 0.28) "GRPN602: M4 to M4 notch (logo) >= 0.280 um.")
\o executing: drc(m5_logo (sep < 0.28) "GRPN602: M5 to M5 space (logo) >= 0.280 um.")
\o drc(m5_logo (notch < 0.28) "GRPN602: M5 to M5 notch (logo) >= 0.280 um.")
\o executing: drc(m6_logo (sep < 0.28) "GRPN602: M6 to M6 space (logo) >= 0.280 um.")
\o drc(m6_logo (notch < 0.28) "GRPN602: M6 to M6 notch (logo) >= 0.280 um.")
\o executing: errpn907_dv = geomAnd(dv logobnd)
\o executing: errpn907_tv = geomAnd(tv logobnd)
\o executing: errpn907_fv = geomAnd(fv logobnd)
\o executing: errpn907_tvdummy = geomAnd(tvdummy logobnd)
\o executing: saveDerived(errpn907_dv "GRPN907: DV over LOGOBND found!")
\o executing: saveDerived(errpn907_tv "GRPN907: TV over LOGOBND found!")
\o executing: saveDerived(errpn907_fv "GRPN907: FV over LOGOBND found!")
\o executing: saveDerived(errpn907_tvdummy "GRPN907: TVDUMMY over LOGOBND found!")
\o DRC started.....Sat Nov 26 13:54:54 2005
\o completed....Sat Nov 26 13:56:53 2005
\o CPU TIME = 00:01:07 TOTAL TIME = 00:01:59
\o ********* Summary of rule violations for cell "MyVerilogCell layout" *********
\o Total errors found: 0
\o
\r t
\r t
\a hiResizeWindow(window(1) list(278:0 1003:933))
\r t
\a hiResizeWindow(window(1) list(278:218 1009:933))

7.3 LVS Report

@(#)$CDS: LVS.exe version 5.0.0 08/24/2005 19:50 (cds12107) $

Command line:
/home/cad/cadence_new/ic/5.0.33/tools/dfII/bin/32bit/LVS.exe -dir /home/002/s/sc/scm042000/cad/cadence/LVS -l -s -t /home/002/s/sc/scm042000/cad/cadence/LVS/layout /home/002/s/sc/scm042000/cad/cadence/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.

Net-list summary for /home/002/s/sc/scm042000/cad/cadence/LVS/layout/netlist

count	
558	nets
8	terminals
538	nfet
538	pfet

Net-list summary for
/home/002/s/sc/scm042000/cad/cadence/LVS/schematic/netlist
count

558	nets
8	terminals
538	nfet
538	pfet

Terminal correspondence points

N504	N112	clk
N414	N1	gnd!
N182	N109	in1
N77	N120	in2
N179	N52	out1
N207	N88	out2
N348	N99	reset
N1	N0	vdd!

Devices in the netlist but not in the rules:
pcapacitor
Devices in the rules but not in the netlist:
subc lvsres dgnfet dgnfetm dgnfet_rf dgnfet_rfm dgnfettw dgnfettwm
dgnfettw_rf dgnfettw_rfm dgpfet dgpfetm dgpfet_rf dgpfet_rfm lpnfet
lpnfetm lpnfet_rf lpnfet_rfm lppfet lppfetm lppfet_rf lppfet_rfm nfetm
nfet33 nfet33m nfet33_rf nfet33_rfm nfet_rf nfet_rfm nfettw nfettwm
nfettw_rf nfettw_rfm pfetm pfet33 pfet33m pfet33_rf pfet33_rfm pfet_rf
pfet_rfm zvtdgnfet zvtdgnfetm zvtdgnfet_rf zvtdgnfet_rfm zvtnfet
zvtnfetm zvtnfet_rf zvtnfet_rfm dgncap dgncapm ncap ncapm ind indm
indp indpm inds indsm indline indlinem havar havarm esdvpnp esdvpnpm
dualmimcap dualmimcapm lmmimcap lmmimcapm mimcap mimcapm vpp vppm
l1res l1resm opndres opndresm opppcres opppcresm oprrpres oprrpresm
sblkndres sblkndresm bondpad bondpadm coupledcpw coupledcpwm
coupledwires coupledwiresm singlecpw singlecpwm singlewire singlewirem
esdndsx esdndsxm dipdnw dipdnwm efuse efusem

The net-lists match logically but have mismatched parameters.

	layout	schematic
	instances	
un-matched	0	0
rewired	0	0
size errors	1076	1076
pruned	0	0
active	1076	1076
total	1076	1076
	nets	
un-matched	0	0
merged	0	0
pruned	0	0
active	558	558
total	558	558
	terminals	
un-matched	0	0
matched but different type	0	0
total	8	8

Probe files from
/home/002/s/sc/scm042000/cad/cadence/LVS/schematic

devbad.out:
netbad.out:
mergenet.out:
termbad.out:
prunenet.out:
prunedev.out:
audit.out:

The no. of lines exceeded than specified by the variable lvsLimitLinesInOutFile.

To see the complete information please see the file:
/home/002/s/sc/scm042000/cad/cadence/LVS/schematic/audit.out
Probe files from
/home/002/s/sc/scm042000/cad/cadence/LVS/layout

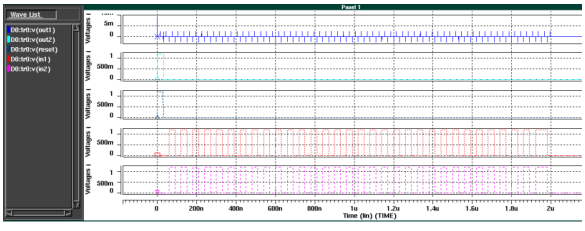
devbad.out:
netbad.out:
mergenet.out:
termbad.out:
prunenet.out:
prunedev.out:

audit.out:
The no. of lines exceeded than specified by the variable lvsLimitLinesInOutFile.

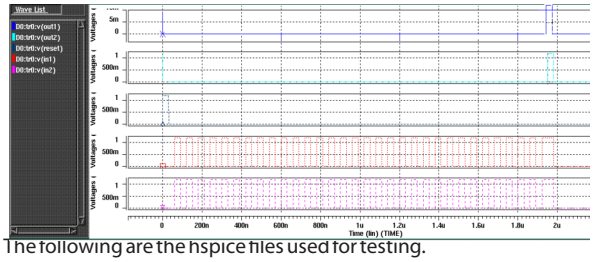
To see the complete information please see the file:
/home/002/s/sc/scm042000/cad/cadence/LVS/layout/audit.out

7. Testing using hspice & Results

The testing of the circuit is done using hspice by feeding all the vectors. The vectors are prepared from the test bench which was used to test the mapped verilog file after synthesis. Unfortunately we didn't get the results. We spent some time debugging and finally we decided to put a break on it at the moment and continue working on this during Fall break. We wonder there could be some set of tools which can help us to map the cells from schematic to the nets in netlist. This will help us to track the state machine, verify the states and give us a good starting point to continue debugging and direction to proceeds either towards inputs or towards outputs. The following are the output waveforms we obtained.



As mentioned earlier the expected output is different from the above. The following is the expected output. We the output high indicates the detection of the Ethernet Frame.



The following are the hspice files used for testing.

```
Final_hspice.sp file

$a sample hspice file
.include "/home/cad/kits/IBM_CMRF8SF-LM013/IBM_PDK/cmrf8sf/relLM/HSPICE/models/model013.lib_inc
"
.include final_netlist
.VEC vector
.option post

vdd vdd gnd 1.2V

Cout_0 out1 gnd 20f
Cout_1 out2 gnd 20f

.tr 1ns 2400ns uic
.end
Vector file
radix 1 1 1 1
vname reset clk in1 in2
slope 0.06
vih 1.2
vil 0
period 15ns
1 0 0 0
1 1 0 0
0 0 0 0
0 1 0 0

0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0

0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0

0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0

0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0

0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0

0 0 1 1
0 1 1 1
0 0 0 0
0 1 0 0
0 0 1 1
0 1 1 1
0 0 0 0
```

```
0100
0011
0111
0000
0100
0011
0111
0000
0100

0011
0111
0000
0100
0011
0111
0000
0100
0011
0111
0011
0111
0000
0100
```

8. Testing using Pathmill

The critical path is from clk to out2 barring reset. The critical path is bolded in the following log from Pathmill.

```
                PathMill (R)
        Version 2003.12 for sparcOS5 -- Nov 15, 2003
        Copyright (c) 1988-2003 by Synopsys, Inc.
                ALL RIGHTS RESERVED

This program is proprietary and confidential information of
Synopsys, Inc.

and may be used and disclosed only as authorized in a license
agreement
controlling such use and disclosure.

Critical Path Analysis Report, Done on Fri Nov 25 03:12:02 2005
CRITICAL PATH SEARCH  - running on 558 nodes
Critical path search completed.

Total number of long paths found     = 8
Total stages traversed = 7282
Total stages computed = 1016

            Process:    typ
            Voltage:    5 volt
            Temperature:            25 degree C

Concise report of longest paths ( Sorted with offset ):

   path   total  # of
   delay  delay stages   from node  =>  to node
=============================================
=================================
 1  0.159  0.159   5     reset (F)     out2 (F)
 2  0.153  0.153   5     reset (F)     out1 (F)
 3  0.091  0.091   6      clk (R)      out2 (F)
 4  0.085  0.085   6      clk (R)      out1 (F)
 5  0.078  0.078   5      clk (F)      out2 (F)
 6  0.076  0.076   5      clk (F)      out2 (R)
 7  0.074  0.074   5      clk (F)      out1 (R)
 8  0.072  0.072   5      clk (F)      out1 (F)

Domino edge reference: &&
                        SP - start of precharge/predischarge phase
                        EP - end of precharge/predischarge phase
                        SE - start of evaluate phase
                        EE - end of evaluate phase

Node Type Index:
                        (C) : Clock node
                        (S) : Sink  node
                        (SZ): Sink  node and turn off edge
                        (A) : Adjusted Latch node delay
                        (a) :  adjusted  Precharge/Predischarge  node
delay
                        (L) : Latch node
                        (F) : Clocked loop node
                        (G) : Gated Clock
                        (T) : Transparent Gated Clock
                        (M) : internal node of timing model
                        (P1): Predischarge node of D1 P-domino
                        (P2): Predischarge node of D2 P-domino
                        (p1): input of D1 P-domino
                        (p2): input of D2 P-domino
                        (N1): precharge node of D1 N-domino
                        (N2): precharge node of D2 N-domino
                        (N3): precharge node of D1 N-domino retain
                        (N4): precharge node of D2 N-domino retain
                        (N5): precharge node of D1 N-domino latch
                        (N6): precharge node of D2 N-domino latch
                        (N7): precharge node of D1 N-domino flop
                        (N8): precharge node of D2 N-domino flop
                        (n1): input of D1 N-domino
                        (n2): input of D2 N-domino
                        (n3): input of D1 N-domino retain
                        (n4): input of D2 N-domino retain
                        (n5): input of D1 N-domino latch
                        (n6): input of D2 N-domino latch
                        (n7): input of D1 N-domino flop
                        (n8): input of D2 N-domino flop
                        (E) : turn off enable edge
                        (Z) : turn off edge

                *** Longest Paths ***

Path (1) : Signal propagates through 5 stages :

 Delay[ns]    R/F          Node            Element
 Acc   Delta time[ns] Cap[pf]&&(Type) Name          Name
 ----- ----- -------  ------  -----------      --------------
 0.000 0.000 0.060 F  0.003     reset
 0.075 0.075 0.127 R  0.037     n113           mx1061
               0.001     n311          mx459
 0.117 0.042 0.034 F  0.003     n307
 0.133 0.016 0.020 R  0.002     n326           mx1013
 0.159 0.026 0.044 F  0.008   (S) out2          mx492
 ------------
 0.159 0.159(total without offset)


Path (2) : Signal propagates through 5 stages :

 Delay[ns]    R/F          Node            Element
 Acc   Delta time[ns] Cap[pf]&&(Type) Name          Name
 ----- ----- -------  ------  -----------      --------------
 0.000 0.000 0.060 F  0.003     reset
 0.075 0.075 0.127 R  0.037     n113           mx1061
               0.001     n218          mx315
```

0.117 0.042 0.034 F 0.003 n215
0.135 0.018 0.025 R 0.003 n232 mx863
0.153 0.018 0.027 F 0.005 (S) out1 mx338

0.153 0.153(total without offset)

Path (3) : Signal propagates through 6 stages :

Delay[ns] Acc	Delta	time[ns]	R/F	Cap[pf]&&(Type)	Node Name	Element Name
0.000	0.000	0.060	R	0.023	clk	
0.021	0.021	0.019	F	0.003	n275	mx404
0.036	0.015	0.023	R	0.003	n280	mx946
0.046	0.010	0.044	F	0.003	n307	mx458
0.065	0.019	0.020	R	0.002	n326	mx1013
0.091	0.026	0.044	F	0.008	(S) out2	mx492

0.091 0.091(total without offset)

Path (4) : Signal propagates through 6 stages :

Delay[ns] Acc	Delta	time[ns]	R/F	Cap[pf]&&(Type)	Node Name	Element Name
0.000	0.000	0.060	R	0.023	clk	
0.021	0.021	0.019	F	0.003	n183	mx262
0.036	0.015	0.023	R	0.003	n186	mx805
0.046	0.010	0.044	F	0.003	n215	mx309
0.067	0.021	0.025	R	0.003	n232	mx863
0.085	0.018	0.027	F	0.005	(S) out1	mx338

0.085 0.085(total without offset)

Path (5) : Signal propagates through 5 stages :

Delay[ns] Acc	Delta	time[ns]	R/F	Cap[pf]&&(Type)	Node Name	Element Name
0.000	0.000	0.060	F	0.023	clk	
0.026	0.026	0.029	R	0.003	n275	mx940
0.036	0.010	0.035	F	0.003	n307	mx448
0.052	0.016	0.020	R	0.002	n326	mx1013
0.078	0.026	0.044	F	0.008	(S) out2	mx492

0.078 0.078(total without offset)

Path (6) : Signal propagates through 5 stages :

Delay[ns] Acc	Delta	time[ns]	R/F	Cap[pf]&&(Type)	Node Name	Element Name
0.000	0.000	0.060	F	0.023	clk	
0.026	0.026	0.029	R	0.003	n275	mx940
0.041	0.015	0.071	R	0.003	n307	mx448
0.061	0.020	0.013	F	0.002	n326	mx475
0.076	0.015	0.025	R	0.008	(S) out2	mx1028

0.076 0.076(total without offset)

Path (7) : Signal propagates through 5 stages :

Delay[ns] Acc	Delta	time[ns]	R/F	Cap[pf]&&(Type)	Node Name	Element Name
0.000	0.000	0.060	F	0.023	clk	
0.026	0.026	0.029	R	0.003	n183	mx800
0.041	0.015	0.071	R	0.003	n215	mx304
0.063	0.022	0.016	F	0.003	n232	mx327
0.074	0.011	0.015	R	0.005	(S) out1	mx874

0.074 0.074(total without offset)

Path (8) : Signal propagates through 5 stages :

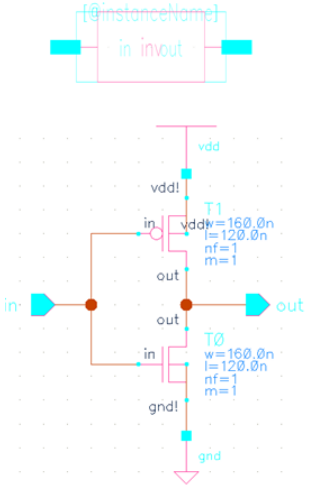
Delay[ns] Acc	Delta	time[ns]	R/F	Cap[pf]&&(Type)	Node Name	Element Name
0.000	0.000	0.060	F	0.023	clk	
0.026	0.026	0.029	R	0.003	n183	mx800
0.036	0.010	0.035	F	0.003	n215	mx304
0.054	0.018	0.025	R	0.003	n232	mx863
0.072	0.018	0.027	F	0.005	(S) out1	mx338

0.072 0.072(total without offset)

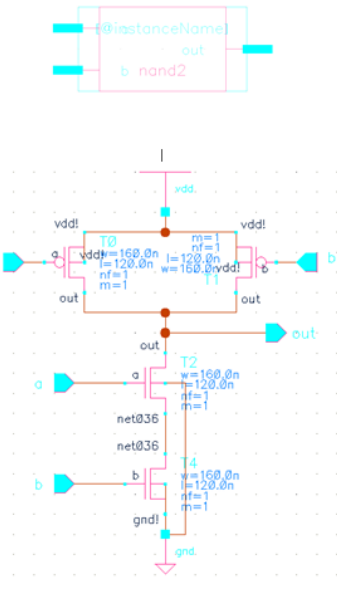
9. Appendix A – Schematics of individual cells

We developed the basic cells during project 4. These are used as fundamental blocks to develop the final cell of interest, Ethernet Frame detector. The following are the symbol and schematic views of the cells.

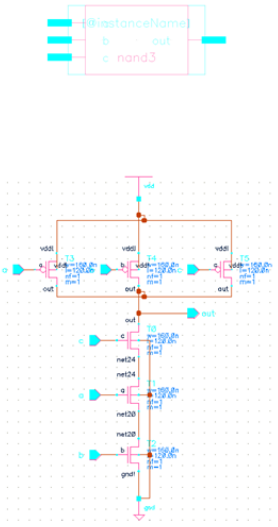
11.1 Inverter



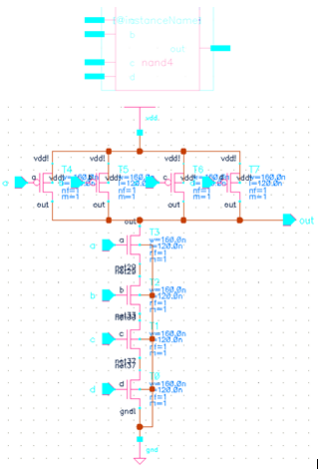
11.2 nand2



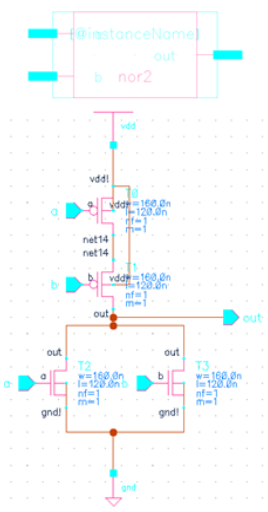
11.3 nand3



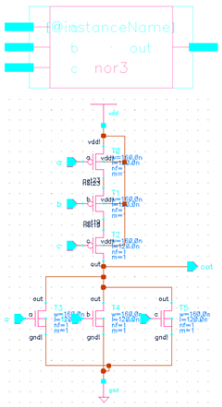
11.4 nand4



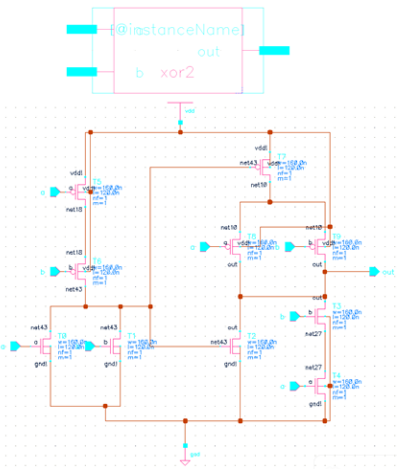
11.5 nor2



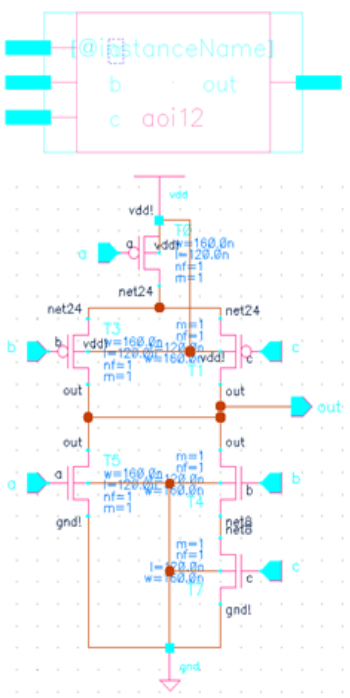
11.6 nor3



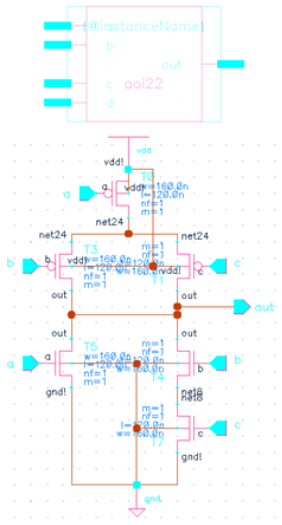
11.7 xor2



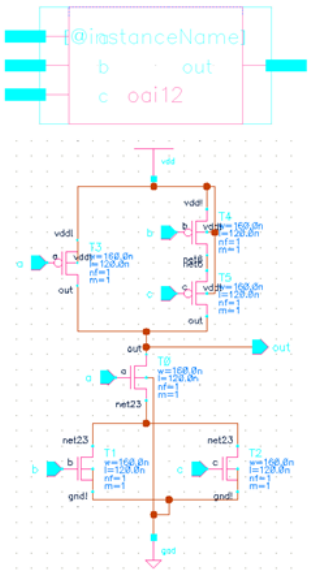
11.8 aoi12



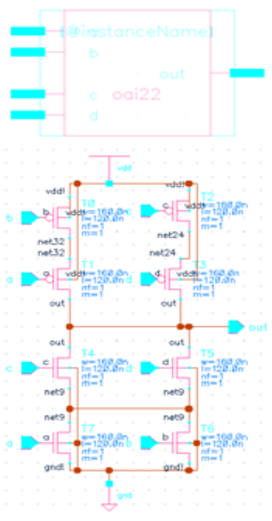
11.9 aoi22



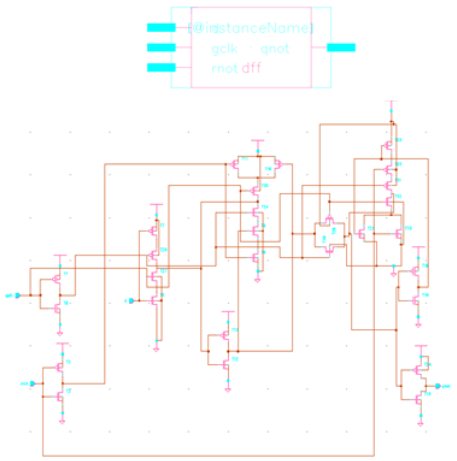
11.10 oai12



11.11 oai22



11.12 dff



10. Appendix B – Verilog files of the Multi Ethernet Frame Detector and Test Bench

```
module MultiEthDetector ( in1, in2, clk, reset, out1, out2 );
 input in1;
 input in2;
 input clk;
 input reset;
 output out1;
 output out2;
   wire    \__tmp52/N389 , \__tmp52/N421 , \__tmp52/N432 , \__tmp52/N441 ,
            \__tmp52/N448 , \__tmp52/N456 , \__tmp51/N389 , \__tmp51/N421 ,
            \__tmp51/N432 , \__tmp51/N441 , \__tmp51/N448 , \__tmp51/N456 , n16,
      n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30,
      n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44,
      n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58,
      n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72,
      n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86,
      n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100,
       n101, n102, n103, n104, n105, n106, n107, n108, n109, n110, n111,
       n112, n113, n114, n115, n116, n117, n118, n119, n120, n121, n122,
       n123, n124, n125, n126, n127, n128, n129, n130, n131, n132, n133,
       n134, n135, n136, n137, n138, n139, n140, n141, n142, n143, n144,
       n145, n146, n147, n148, n149, n150, n151, n152, n153, n154, n155,
     n156, n157, n158;
 assign \__tmp52/N389 = in2;
 assign \__tmp51/N389 = in1;

 inv U15 ( .in(reset), .out(n142) );
 nor3 U16 ( .a(n16), .b(n153), .c(n155), .out(n146) );
 nand3 U17 ( .a(n17), .b(n18), .c(n19), .out(n16) );
 nand2 U18 ( .a(n156), .b(n20), .out(n18) );
 nand3 U19 ( .a(n158), .b(out2), .c(n157), .out(n20) );
 inv U20 ( .in(n144), .out(out2) );
 oai12 U21 ( .a(n23), .b(n21), .c(n22), .out(n17) );
 nand4 U22 ( .a(n24), .b(n25), .c(n26), .d(n27), .out(\__tmp52/N456 )
);
```

```
 oai12 U23 ( .a(n22), .b(n28), .c(n29), .out(n27) );
 nor2 U24 ( .a(n19), .b(n30), .out(n29) );
 inv U25 ( .in(n31), .out(n28) );
 aoi22 U26 ( .a(n32), .b(n33), .c(n34), .d(n35), .out(n26) );
 oai12 U27 ( .a(n30), .b(n36), .c(n37), .out(n33) );
 nand2 U28 ( .a(n157), .b(n35), .out(n30) );
 nand2 U29 ( .a(n38), .b(n22), .out(n24) );
 inv U30 ( .in(n39), .out(n38) );
 nand4 U31 ( .a(n40), .b(n25), .c(n41), .d(n42), .out(\__tmp52/N448 )
);
 nand4 U32 ( .a(n43), .b(n155), .c(n22), .d(n44), .out(n42) );
 aoi22 U33 ( .a(n32), .b(n45), .c(n34), .d(n46), .out(n41) );
 oai22 U34 ( .a(n46), .b(n37), .c(n23), .d(n36), .out(n45) );
 nand4 U35 ( .a(n47), .b(n155), .c(n48), .d(n22), .out(n25) );
 nand2 U36 ( .a(n49), .b(n22), .out(n40) );
 nand3 U37 ( .a(n31), .b(n39), .c(n50), .out(n49) );
 nand2 U38 ( .a(n43), .b(n51), .out(n50) );
 inv U39 ( .in(n36), .out(n43) );
 nand2 U40 ( .a(n158), .b(n46), .out(n36) );
 nand3 U41 ( .a(n47), .b(n48), .c(n52), .out(n39) );
 nand2 U42 ( .a(n48), .b(n51), .out(n31) );
 inv U43 ( .in(n21), .out(n48) );
 nand2 U44 ( .a(n46), .b(n35), .out(n21) );
 inv U45 ( .in(n158), .out(n35) );
 inv U46 ( .in(n157), .out(n46) );
 nand2 U47 ( .a(n53), .b(n54), .out(\__tmp52/N441 ) );
 oai12 U48 ( .a(n22), .b(n55), .c(n56), .out(n54) );
 nor3 U49 ( .a(n57), .b(n153), .c(n156), .out(n56) );
 aoi22 U50 ( .a(n34), .b(n23), .c(n52), .d(n32), .out(n53) );
 nand3 U51 ( .a(n58), .b(n59), .c(n60), .out(\__tmp52/N432 ) );
 aoi22 U52 ( .a(n34), .b(n57), .c(n32), .d(n155), .out(n60) );
 nor2 U53 ( .a(n61), .b(\__tmp52/N389 ), .out(n32) );
 inv U54 ( .in(n47), .out(n61) );
 nor2 U55 ( .a(n153), .b(n154), .out(n47) );
 nor2 U56 ( .a(n22), .b(n44), .out(n34) );
 nand3 U57 ( .a(n51), .b(n22), .c(n52), .out(n59) );
 nor2 U58 ( .a(n23), .b(n155), .out(n52) );
 inv U59 ( .in(n19), .out(n51) );
 nand2 U60 ( .a(n55), .b(n22), .out(n58) );
 inv U61 ( .in(\__tmp52/N389 ), .out(n22) );
 nor2 U62 ( .a(n37), .b(n19), .out(n55) );
 nand2 U63 ( .a(n57), .b(n23), .out(n37) );
 inv U64 ( .in(n156), .out(n23) );
 inv U65 ( .in(n155), .out(n57) );
  oai12 U66 ( .a(n62), .b(\__tmp52/N389 ), .c(n19),
.out(\__tmp52/N421 ) );
 nand3 U67 ( .a(n153), .b(n63), .c(\__tmp52/N389 ), .out(n62) );
 inv U68 ( .in(n154), .out(n63) );
 nand2 U69 ( .a(n154), .b(n44), .out(n19) );
 inv U70 ( .in(n153), .out(n44) );
 oai12 U71 ( .a(n66), .b(n64), .c(n65), .out(n145) );
 nand4 U72 ( .a(\__tmp51/N389 ), .b(n67), .c(n68), .d(n69), .out(n66)
);
 nand4 U73 ( .a(n70), .b(n150), .c(n71), .d(n72), .out(n65) );
 inv U74 ( .in(n73), .out(n72) );
 inv U75 ( .in(n74), .out(n70) );
 nand4 U76 ( .a(n75), .b(out1), .c(n76), .d(n77), .out(n64) );
 inv U77 ( .in(n143), .out(out1) );
 nand4 U78 ( .a(n78), .b(n79), .c(n80), .d(n81), .out(\__tmp51/N456 )
);
 inv U79 ( .in(n82), .out(n81) );
 oai12 U80 ( .a(n84), .b(n83), .c(\__tmp51/N389 ), .out(n82) );
 aoi22 U81 ( .a(n69), .b(n85), .c(n86), .d(n87), .out(n84) );
 nand3 U82 ( .a(n88), .b(n89), .c(n86), .out(n78) );
 nand2 U83 ( .a(n90), .b(n91), .out(\__tmp51/N448 ) );
 aoi22 U84 ( .a(n74), .b(n89), .c(\__tmp51/N389 ), .d(n92), .out(n91) );
 nand3 U85 ( .a(n93), .b(n94), .c(n95), .out(n92) );
 inv U86 ( .in(n96), .out(n94) );
 nand4 U87 ( .a(n83), .b(n97), .c(n98), .d(n99), .out(n74) );
 nor3 U88 ( .a(n100), .b(n101), .c(n102), .out(n99) );
 nand2 U89 ( .a(n103), .b(n77), .out(n97) );
 nand2 U90 ( .a(n104), .b(n67), .out(n83) );
 inv U91 ( .in(n105), .out(n104) );
 aoi22 U92 ( .a(n87), .b(n103), .c(n85), .d(n106), .out(n90) );
 nor2 U93 ( .a(n107), .b(n151), .out(n103) );
  nand4 U94 ( .a(n79), .b(n108), .c(n109), .d(n110),
.out(\__tmp51/N441 ) );
 nor3 U95 ( .a(n111), .b(n112), .c(n113), .out(n110) );
  oai22 U96 ( .a(\__tmp51/N389 ), .b(n114), .c(n115), .d(n89),
.out(n113) );
 aoi12 U97 ( .a(n117), .b(n116), .c(n106), .out(n115) );
 inv U98 ( .in(n93), .out(n117) );
 nand2 U99 ( .a(n116), .b(n69), .out(n93) );
 aoi12 U100 ( .a(n100), .b(n106), .c(n118), .out(n114) );
 nor2 U101 ( .a(n75), .b(n119), .out(n100) );
 inv U102 ( .in(n69), .out(n119) );
 inv U103 ( .in(n118), .out(n75) );
 inv U104 ( .in(n120), .out(n112) );
  aoi22 U105 ( .a(n89), .b(n121), .c(\__tmp51/N389 ), .d(n96),
.out(n120) );
 nor2 U106 ( .a(n77), .b(n122), .out(n96) );
 oai12 U107 ( .a(n98), .b(n123), .c(n71), .out(n121) );
 nand2 U108 ( .a(n124), .b(n69), .out(n98) );
  oai22 U109 ( .a(\__tmp51/N389 ), .b(n125), .c(\__tmp51/N389 ),
.d(n126),
    .out(n111) );
 aoi12 U110 ( .a(n101), .b(n127), .c(n88), .out(n126) );
 nor2 U111 ( .a(n122), .b(n128), .out(n101) );
 aoi12 U112 ( .a(n102), .b(n127), .c(n67), .out(n125) );
 nor2 U113 ( .a(n122), .b(n129), .out(n102) );
 nand2 U114 ( .a(n149), .b(n69), .out(n122) );
 aoi22 U115 ( .a(n127), .b(n87), .c(n85), .d(n150), .out(n109) );
 nor3 U116 ( .a(n129), .b(\__tmp51/N389 ), .c(n130), .out(n85) );
 nor2 U117 ( .a(n77), .b(n89), .out(n87) );
 nor2 U118 ( .a(n123), .b(n76), .out(n127) );
 nand2 U119 ( .a(n73), .b(n89), .out(n79) );
 nor2 U120 ( .a(n131), .b(n129), .out(n73) );
 nand3 U121 ( .a(n80), .b(n132), .c(n133), .out(\__tmp51/N432 ) );
  aoi22 U122 ( .a(n116), .b(\__tmp51/N389 ), .c(n118), .d(n89),
.out(n133) );
 nor2 U123 ( .a(n128), .b(n130), .out(n118) );
 nor2 U124 ( .a(n77), .b(n130), .out(n116) );
 nand2 U125 ( .a(n152), .b(n76), .out(n130) );
 nand3 U126 ( .a(n67), .b(n89), .c(n149), .out(n132) );
 inv U127 ( .in(n134), .out(n80) );
 nand4 U128 ( .a(n135), .b(n108), .c(n136), .d(n137), .out(n134) );
 nand3 U129 ( .a(n147), .b(n138), .c(\__tmp51/N389 ), .out(n137) );
 inv U130 ( .in(n131), .out(n138) );
 nand2 U131 ( .a(n68), .b(n150), .out(n131) );
 nand2 U132 ( .a(n124), .b(n89), .out(n136) );
 inv U133 ( .in(n71), .out(n124) );
 nand2 U134 ( .a(n88), .b(n68), .out(n71) );
 inv U135 ( .in(n128), .out(n88) );
 nand2 U136 ( .a(\__tmp51/N389 ), .b(n139), .out(n108) );
 oai12 U137 ( .a(n95), .b(n105), .c(n77), .out(n139) );
 nand3 U138 ( .a(n68), .b(n69), .c(n147), .out(n95) );
 nor2 U139 ( .a(n150), .b(n151), .out(n69) );
 nand2 U140 ( .a(n106), .b(n68), .out(n105) );
 nor2 U141 ( .a(n149), .b(n152), .out(n68) );
 inv U142 ( .in(n123), .out(n106) );
 nand2 U143 ( .a(n151), .b(n107), .out(n123) );
 inv U144 ( .in(n150), .out(n107) );
 nand3 U145 ( .a(n67), .b(n89), .c(n86), .out(n135) );
 nor2 U146 ( .a(n76), .b(n152), .out(n86) );
 inv U147 ( .in(n149), .out(n76) );
 inv U148 ( .in(\__tmp51/N389 ), .out(n89) );
 inv U149 ( .in(n129), .out(n67) );
 nand2 U150 ( .a(n77), .b(n140), .out(n129) );
```

```
   oai12 U151 ( .a(n141), .b(\__tmp51/N389 ), .c(n128),
.out(\__tmp51/N421 ));
 nand3 U152 ( .a(n147), .b(n140), .c(\__tmp51/N389 ), .out(n141) );
 inv U153 ( .in(n148), .out(n140) );
 nand2 U154 ( .a(n148), .b(n77), .out(n128) );
 inv U155 ( .in(n147), .out(n77) );
 dff \__tmp51/z_reg ( .d(n145), .gclk(clk), .rnot(n142), .qnot(n143) );
 dff \__tmp52/z_reg ( .d(n146), .gclk(clk), .rnot(n142), .qnot(n144) );
   dff \__tmp51/state_reg<0> ( .d(\__tmp51/N389 ), .gclk(clk),
.rnot(n142),
     .qnot(n147) );
   dff \__tmp51/state_reg<1> ( .d(\__tmp51/N421 ), .gclk(clk),
.rnot(n142),
     .qnot(n148) );
   dff \__tmp51/state_reg<2> ( .d(\__tmp51/N432 ), .gclk(clk),
.rnot(n142),
     .qnot(n149) );
   dff \__tmp51/state_reg<3> ( .d(\__tmp51/N441 ), .gclk(clk),
.rnot(n142),
     .qnot(n150) );
   dff \__tmp51/state_reg<4> ( .d(\__tmp51/N448 ), .gclk(clk),
.rnot(n142),
     .qnot(n151) );
   dff \__tmp51/state_reg<5> ( .d(\__tmp51/N456 ), .gclk(clk),
.rnot(n142),
     .qnot(n152) );
   dff \__tmp52/state_reg<0> ( .d(\__tmp52/N389 ), .gclk(clk),
.rnot(n142),
     .qnot(n153) );
   dff \__tmp52/state_reg<1> ( .d(\__tmp52/N421 ), .gclk(clk),
.rnot(n142),
     .qnot(n154) );
   dff \__tmp52/state_reg<2> ( .d(\__tmp52/N432 ), .gclk(clk),
.rnot(n142),
     .qnot(n155) );
   dff \__tmp52/state_reg<3> ( .d(\__tmp52/N441 ), .gclk(clk),
.rnot(n142),
     .qnot(n156) );
   dff \__tmp52/state_reg<4> ( .d(\__tmp52/N448 ), .gclk(clk),
.rnot(n142),
     .qnot(n157) );
   dff \__tmp52/state_reg<5> ( .d(\__tmp52/N456 ), .gclk(clk),
.rnot(n142),
     .qnot(n158) );
Endmodule

//Testbench of above code

module testSequeneDetector ();
  reg in1, in2, CLK, reset;
  wire out1, out2;

 MultiEthDetector mySD (in1, in2, CLK, reset, out1, out2);
 //initial
 //begin
 //
 //#1 CLK = 0;
 //forever
 //  begin
 //   #30 CLK = ~CLK;
 //  end
 //end

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

initial
begin

CLK=0;
reset = 1;
in1 = 0;
in2 = 0;
#30reset = 0;

#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
```

```
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;
#30 in1 = 0;
  in2 = 0;
#30 in1 = 1;
  in2 = 1;

  //#30 in1 =
  //#30 in1 =
  //#30 in1 =
  //#30 in1 =
  //#30 in1 =
  //#30 in1 =
  //#30 in1 =
  #30 $finish;
end
always #15 CLK = ~CLK;
endmodule
```

11. Conclusions

We learnt the whole process of the layout of a basic cell to layout and routing of the entire circuit. Went through the IBM 0.13 micron process to lay out the cells and got a good understanding of Design Rules, the tool being Cadence. We used synopsys to synthesize the circuit. Laid out and routed the cells using Encounter. The Off grid errors shown by Encounter is often annoying and we don't understand why the tool throws these errors on the first pace because it is the tool which routed the circuit.

Even though we went through all the tools, most of it is following blindly from the tutorial. We tried to explore the meaning of this by Web pages as and when possible. If any thing goes wrong, we have no means to understand as no Reference documents were provided. It would greatly help the students if documentation references are provided for all the tools we use.

Acknowledgements

We would like to thank the professor Dr. Carl Sachen for co-operation and extended help.