The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

RTL-to-GDSII Implementation of a Universal Asynchronous Receiver-Transmitter (UART) in SCL

Pandala Charishma, Dr. Mohd Ziauddin Jahangir, Dr. D Krishna Reddy
VLSID, Dept of ECE, Chaitanya Bharathi Institute of Technology,Hyderabad, Telangana-India
Corresponding Author: charishmapandala2025@gmail.com

SCL లో యూనివర్సల్ ఎసిన్క్రోనస్ రిసీవర్-ట్రాన్స్మిటర్ (UART) యొక్క RTL-to-GDSII అమలు

పండాల చరిష్మా, డా. మొహమ్మద్ జియావుద్దీన్ జహంగీర్, డా. డి కృష్ణా రెడ్డి
VLSID, ECE విభాగం, చైతన్య భారతి ఇన్స్టిట్యూట్ ఆఫ్ టెక్నాలజీ, హైదరాబాద్, తెలంగాణ-భారతదేశం
సంబంధిత రచయిత : charishmapandala2025@gmail.com

వియుక్త

ఈ పత్రం SCL PDK ని ఉపయోగించి RTL నుండి GDS II కి యూనివర్సల్ అసమకాలిక రిసీవర్-ట్రాన్స్మిటర్ (UART) యొక్క పూర్తి డిజిటల్ ప్రవాహ అమలును అందిస్తుంది. UART అనేది ఎంబెడెడ్ మరియు VLSI వ్యవస్థలలో విస్తృతంగా ఉపయోగించే సీరియల్ కమ్యూనికేషన్ ప్రోటోకాల్‌లలో ఒకటి. ప్రతిపాదిత UART డిజైన్‌లో ట్రాన్స్మిటర్, రిసీవర్ మరియు బాడ్ రేట్ జనరేటర్ ఉంటాయి. ఈ డిజైన్ వెరిలాగ్‌లో మోడల్ చేయబడింది, సిమ్యులేషన్ ద్వారా క్రియాత్మకంగా ధృవీకరించబడింది, సంశ్లేషణ చేయబడింది మరియు కాడెన్స్ సాధనాలలో పూర్తి RTL-to-GDSII డిజిటల్ ప్రవాహాన్ని ఉపయోగించి అమలు చేయబడింది. పోస్ట్-లేఅవుట్ ఫలితాలు విజయవంతమైన టైమింగ్ క్లోజర్ మరియు తయారు చేయగల GDSII ఉత్పత్తిని నిర్ధారిస్తాయి, ASIC మరియు FPGA అప్లికేషన్‌ల కోసం డిజైన్ యొక్క అనుకూలతను ప్రదర్శిస్తాయి. పూర్తి ప్రవాహం CADENCE సాధనాలను ఉపయోగించి అమలు చేయబడుతుంది మరియు భౌతిక ధృవీకరణ సీమెన్స్ CALIBRE ని ఉపయోగించి నిర్వహించబడుతుంది.

కీలకపదాలు

UART, డిజిటల్ డిజైన్ ఫ్లో, RTL -టు- GDSII, వెరిలాగ్ HDL, ASIC, FPGA

I . పరిచయం

ఆధునిక డిజిటల్ వ్యవస్థలలో సీరియల్ కమ్యూనికేషన్ చాలా అవసరం, ప్రాసెసర్లు, పెరిఫెరల్స్ మరియు బాహ్య పరికరాల మధ్య డేటా బదిలీని అనుమతిస్తుంది [3]. అందుబాటులో ఉన్న ప్రోటోకాల్‌లలో, UART దాని సరళత, తక్కువ పిన్ కౌంట్ మరియు అమలు సౌలభ్యం కారణంగా విస్తృతంగా స్వీకరించబడింది [4].

ఈ పత్రం UART ని కేస్ స్టడీగా ఉపయోగించి పూర్తి డిజిటల్ డిజైన్ ప్రవాహాన్ని (RTL నుండి GDSII) ప్రదర్శిస్తుంది. పరిశ్రమ-ప్రామాణిక CAD సాధనాలను వర్తింపజేయడం ద్వారా, డిజైన్ RTL మోడలింగ్, ఫంక్షనల్ సిమ్యులేషన్, లాజిక్ సింథసిస్, ఫ్లోర్‌ప్లానింగ్, ప్లేస్‌మెంట్ మరియు రూటింగ్, స్టాటిక్ టైమింగ్ అనాలిసిస్ (STA) మరియు GDSII జనరేషన్‌తో సహా అన్ని దశల ద్వారా నిర్వహించబడుతుంది.

II. నేపథ్యం మరియు సంబంధిత పని

UART కమ్యూనికేషన్ అనేది స్టార్ట్, డేటా, పారిటీ (ఐచ్ఛికం) మరియు స్టాప్ బిట్‌లతో సీరియల్ డేటాను ప్రసారం చేయడం మరియు స్వీకరించడం. మా పని పూర్తి RTL-to-GDSII అమలును నొక్కి చెబుతుంది, సాధారణంగా ఉపయోగించే కమ్యూనికేషన్ బ్లాక్‌ను ప్రవర్తనా వివరణ నుండి లేఅవుట్‌కు ఎలా తీసుకోవచ్చో హైలైట్ చేస్తుంది.

III. పద్ధతి

UART డిజైన్ అభివృద్ధి ప్రామాణిక RTL-to-GDSII డిజిటల్ డిజైన్ ప్రవాహాన్ని ఉపయోగించి జరిగింది. ఈ పద్ధతిలోని ప్రతి దశ హార్డ్‌వేర్ వివరణను ఒక వియుక్త ప్రవర్తనా నమూనా నుండి తయారు చేయగల లేఅవుట్‌కు క్రమంగా మెరుగుపరుస్తుంది. ఈ ప్రవాహాన్ని ఈ క్రింది ప్రధాన దశలుగా విభజించవచ్చు:

1. రిజిస్టర్ బదిలీ స్థాయి (RTL) డిజైన్

ప్రారంభ దశలో, UART హార్డ్‌వేర్ డిస్క్రిప్షన్ లాంగ్వేజ్ (HDL)లో, ప్రత్యేకంగా వెరిలాగ్‌లో వివరించబడింది. RTL మోడల్ ట్రాన్స్‌మిటర్, రిసీవర్ మరియు బాడ్ రేట్ జనరేటర్ వంటి మాడ్యూల్‌లతో సహా UART యొక్క క్రియాత్మక ప్రవర్తన మరియు సమయాన్ని సంగ్రహిస్తుంది. RTL అనేది గోల్డెన్ రిఫరెన్స్ డిజైన్‌గా పనిచేస్తుంది. ఈ దశలో ఇన్‌పుట్‌లలో వెరిలాగ్ సోర్స్ కోడ్, ప్రాజెక్ట్ కాన్ఫిగరేషన్ ఫైల్ మరియు ఫంక్షనల్ వెరిఫికేషన్ కోసం సిమ్యులేషన్ టెస్ట్‌బెంచ్ ఉన్నాయి. అవుట్‌పుట్ అనేది ధృవీకరించబడిన RTL మోడల్, దీనిని గేట్-లెవల్ వివరణలో సంశ్లేషణ చేయవచ్చు.

2. RTL సిమ్యులేషన్ మరియు ధృవీకరణ

UART యొక్క వెరిలాగ్ వివరణను వ్రాసిన తర్వాత మొదటి దశ ఫంక్షనల్ సిమ్యులేషన్. ఈ దశ సంశ్లేషణకు వెళ్ళే ముందు డిజైన్ ప్రవర్తన ఉద్దేశించిన UART ప్రోటోకాల్ స్పెసిఫికేషన్‌కు సరిపోలుతుందని నిర్ధారిస్తుంది. ఇన్‌పుట్ బస్ (డేటా[7:0])లో ట్రాన్స్‌మిషన్‌ను ప్రారంభించడం మరియు డేటా పదాలను అందించడం ద్వారా డిజైన్‌కు ఉద్దీపనను వర్తింపజేయడానికి ఒక టెస్ట్‌బెంచ్ అభివృద్ధి చేయబడింది. నియంత్రణ సిగ్నల్‌లతో (rd y, rd y_clr, మరియు tx_busy) పాటు అవుట్‌పుట్ బస్ (rxdata [7:0])ను కూడా అనుకరణ పర్యవేక్షించింది.

Figure 1 లో చూపబడిన సిమ్యులేషన్ వేవ్‌ఫార్మ్, UART యొక్క సరైన ఆపరేషన్‌ను ప్రదర్శిస్తుంది. 00h నుండి 0Fh వరకు ఉన్న డేటా పదాలు ఇన్‌పుట్ బస్‌లో వరుసగా ప్రసారం చేయబడతాయి. రిసీవర్ ఈ పదాలను rxdata అవుట్‌పుట్‌లో విజయవంతంగా పునర్నిర్మిస్తుంది, సరైన ట్రాన్స్‌మిషన్ మరియు రిసెప్షన్‌ను ధృవీకరిస్తుంది. యాక్టివ్ ట్రాన్స్‌మిషన్ సమయంలో tx_busy సిగ్నల్ ఎక్కువగా ఉంటుంది మరియు rd y ఫ్లాగ్ విజయవంతమైన డేటా రిసెప్షన్‌పై నొక్కి చెబుతుంది, ఇది ట్రాన్స్‌మిటర్ మరియు రిసీవర్ మాడ్యూళ్ల మధ్య సరైన సమకాలీకరణను సూచిస్తుంది.

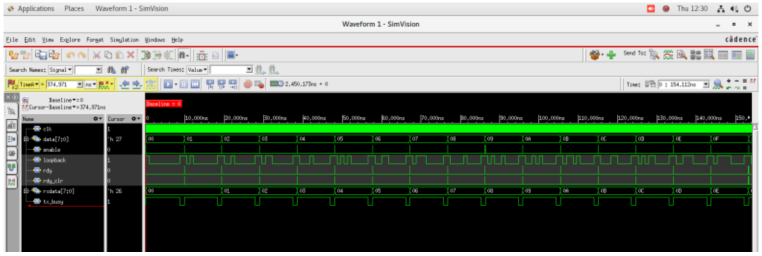

Applications
Places
Waveform 1 - SimVision
Thu 12:30
File Edit View Explore Format Simulation Windows Help
cādence
Search Names: Signal
Search Times: Value
Baseline = 0
Cursor-Baseline = 374,971ns
Name
Cursor
clk
data[7:0]
enable
loopback
rdy
rdy_clr
rxdata[7:0]
tx_busy
'h 27
'h 26
10,000ns
20,000ns
30,000ns
40,000ns
50,000ns
60,000ns
70,000ns
80,000ns
90,000ns
100,000ns
110,000ns
120,000ns
130,000ns
140,000ns

Fig.1 : ఇన్నిసివ్ కాడెన్స్ టూల్‌లో UART అనుకరణ

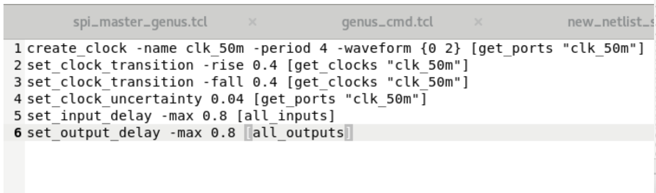
ఈ దశ క్రియాత్మకంగా ధృవీకరించబడిన RTL నమూనాను ఉత్పత్తి చేస్తుంది, ఇది దిగువ అమలు దశలకు బంగారు డిజైన్‌గా పనిచేస్తుంది. ఈ ఖచ్చితత్వాన్ని ధృవీకరించిన తర్వాత మాత్రమే డిజైన్‌ను గేట్-లెవల్ నెట్‌లిస్ట్‌లుగా సంశ్లేషణ చేయవచ్చు.

3. లాజిక్ సంశ్లేషణ

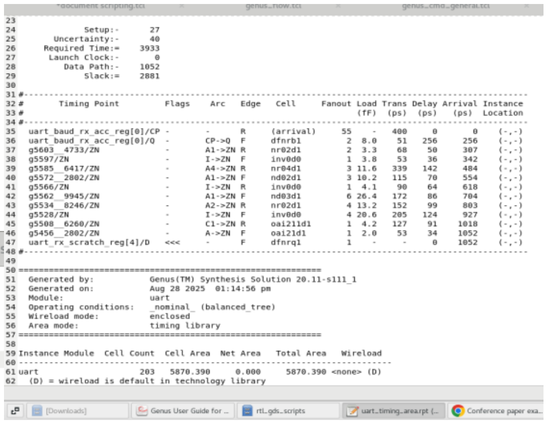
సిమ్యులేషన్ ద్వారా RTL డిజైన్ యొక్క క్రియాత్మక ఖచ్చితత్వం స్థాపించబడిన తర్వాత, UART వివరణను కాడెన్స్ జెనస్ సాధనాన్ని ఉపయోగించి గేట్-లెవల్ నెట్‌లిస్ట్‌గా సంశ్లేషణ చేశారు. సంశ్లేషణ

ప్రవర్తనా వెరిలాగ్‌ను ఎంచుకున్న టెక్నాలజీ నోడ్ యొక్క ప్రామాణిక సెల్ లైబ్రరీపై మ్యాప్ చేయబడిన టెక్నాలజీ-ఆధారిత వివరణగా మారుస్తుంది.

సంశ్లేషణకు ఇన్‌పుట్‌లలో UART RTL , లిబర్టీ (.lib) ఫార్మాట్‌లోని టెక్నాలజీ లైబ్రరీ మరియు సినాప్సిస్ డిజైన్ పరిమితులు (SDC) ఫార్మాట్‌లో నిర్వచించబడిన సమయ మరియు గడియార పరిమితుల సమితి ఉన్నాయి. ఈ పరిమితులు ఆపరేటింగ్ ఫ్రీక్వెన్సీ, ఇన్‌పుట్/అవుట్‌పుట్ ఆలస్యం మరియు డిజైన్ అవసరాలను నిర్దేశిస్తాయి.



```
create_clock -name clk_50m -period 4 -waveform {0 2} [get_ports "clk_50m"]
set_clock_transition -rise 0.4 [get_clocks "clk_50m"]
set_clock_transition -fall 0.4 [get_clocks "clk_50m"]
set_clock_uncertainty 0.04 [get_ports "clk_50m"]
set_input_delay -max 0.8 [all_inputs]
set_output_delay -max 0.8 [all_outputs]
```



```

          Setup:-      27
    Uncertainty:-      40
  Required Time:=    3933
   Launch Clock:-       0
      Data Path:-    1052
          Slack:=    2881

#---------------------------------------------------------------------------------------------------------
#      Timing Point          Flags    Arc    Edge   Cell        Fanout Load Trans Delay Arrival Instance
#                                                                      (fF)  (ps)  (ps)   (ps)  Location
#---------------------------------------------------------------------------------------------------------
  uart_baud_rx_acc_reg[0]/CP -        -      R      (arrival)     55    -   400     0       0    (-,-)
  uart_baud_rx_acc_reg[0]/Q  -        CP->Q  F      dfnrb1         2  8.0    51   256     256    (-,-)
  g5603__4733/ZN             -        A1->ZN R      nr02d1         2  3.3    68    50     307    (-,-)
  g5597/ZN                   -        I->ZN  F      inv0d0         1  3.8    53    36     342    (-,-)
  g5585__6417/ZN             -        A4->ZN R      nr04d1         3 11.6   339   142     484    (-,-)
  g5572__2802/ZN             -        A1->ZN F      nd02d1         3 10.2   115    70     554    (-,-)
  g5566/ZN                   -        I->ZN  R      inv0d0         1  4.1    90    64     618    (-,-)
  g5562__9945/ZN             -        A1->ZN F      nd03d1         6 26.4   172    86     704    (-,-)
  g5534__8246/ZN             -        A2->ZN R      nr02d1         4 13.2   152    99     803    (-,-)
  g5528/ZN                   -        I->ZN  F      inv0d0         4 20.6   205   124     927    (-,-)
  g5508__6260/ZN             -        C1->ZN R      oai211d1       1  4.2   127    91    1018    (-,-)
  g5456__2802/ZN             -        A->ZN  F      oai21d1        1  2.0    53    34    1052    (-,-)
  uart_rx_scratch_reg[4]/D   <<<      -      F      dfnrq1         1    -     -     0    1052    (-,-)
#---------------------------------------------------------------------------------------------------------


===================================================================
  Generated by:            Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:            Aug 28 2025  01:14:56 pm
  Module:                  uart
  Operating conditions:    _nominal_ (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
===================================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------
uart                    203   5870.390     0.000     5870.390 <none> (D)
  (D) = wireload is default in technology library
```

Fig.2: సారాంశం డిజైన్ పరిమితులు ఫైల్ Fig.3: జెనస్ టూల్‌లో సంశ్లేషణ నివేదిక

ఈ దశ యొక్క అవుట్‌పుట్ అనేది ఫ్లిప్-ఫ్లాప్స్, NAND, NOR, మరియు multiplexers.netlist వంటి ప్రామాణిక సెల్‌ల పరంగా పూర్తిగా వ్యక్తీకరించబడిన గేట్-స్థాయి నెట్‌లిస్ట్, ఇది <ddesign_netlist.v రూపంలో ఉంటుంది. అదనంగా, డిజైన్ దాని పనితీరు లక్ష్యాలను చేరుకుంటుందో లేదో అంచనా వేయడానికి వివరణాత్మక సమయం, ప్రాంతం మరియు శక్తి నివేదికలు రూపొందించబడతాయి.

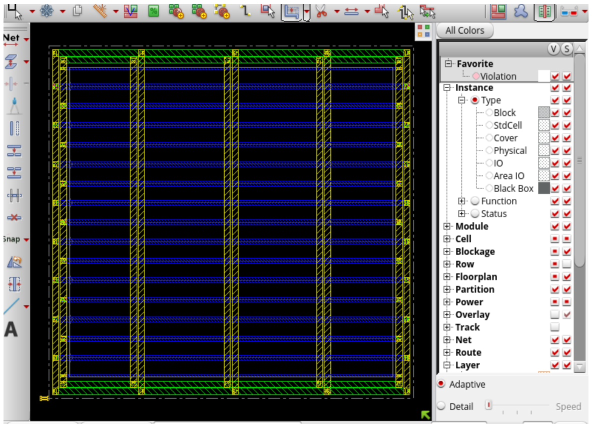
చిత్రం 3లో చూపబడిన సంశ్లేషణ నివేదిక, UART యొక్క క్లిష్టమైన సమయ మార్గాన్ని ప్రదర్శిస్తుంది. డేటాపాత్ ఆలస్యం 1052 ps గా నివేదించబడింది, అయితే అవసరమైన సమయం 3933 ps , ఇది 2881 ps యొక్క సానుకూల స్లాక్‌ను ఇస్తుంది. ఇది డిజైన్ ఉల్లంఘనలు లేకుండా లక్ష్య సమయ నిర్దేశాలను సౌకర్యవంతంగా తీరుస్తుందని సూచిస్తుంది. నివేదిక సెల్ గణన మరియు ప్రాంత వినియోగాన్ని కూడా సంగ్రహిస్తుంది, సంశ్లేషణ చేయబడిన UART 203 ప్రామాణిక కణాలను కలిగి ఉంటుంది మరియు 5870.39 μm^2 ప్రాంతాన్ని ఆక్రమించింది. అందువలన, ఈ దశ చివరిలో, UART సమయం మరియు

ప్రాంత పరిమితులను సంతృప్తిపరిచే గేట్-స్థాయి నెట్‌లిస్ట్‌గా అందుబాటులో ఉంటుంది మరియు ఈ నెట్‌లిస్ట్ భౌతిక రూపకల్పనకు ఇన్‌పుట్‌గా పనిచేస్తుంది.

4. ఫ్లోర్ ప్లానింగ్ మరియు పవర్ ప్లానింగ్

ఫ్లోర్‌ప్లానింగ్ చిప్ యొక్క ప్రారంభ భౌతిక లేఅవుట్‌ను నిర్వచిస్తుంది. సంశ్లేషణ సమయంలో నివేదించబడిన వినియోగం ఆధారంగా డై పరిమాణం అంచనా వేయబడుతుంది మరియు UART కోర్ ప్రాంతం తదనుగుణంగా నిర్వచించబడుతుంది. ఇంటర్‌ఫేస్ అవసరాలను తీర్చడానికి ఇన్‌పుట్/అవుట్‌పుట్ పిన్‌లను అంచు చుట్టూ ఉంచుతారు. ఈ దశలో, బలమైన విద్యుత్ పంపిణీ నెట్‌వర్క్‌ను సృష్టించడానికి పవర్ ప్లానింగ్ కూడా నిర్వహిస్తారు. VDD మరియు VSS కోసం పవర్ రింగులు కోర్ సరిహద్దు చుట్టూ నిర్మించబడతాయి, ప్లేస్‌మెంట్ వరుసలలో క్షితిజ సమాంతర మరియు నిలువు పవర్ రైళ్లను చొప్పించబడతాయి మరియు ఏకరీతి వోల్టేజ్ డెలివరీని నిర్ధారించడానికి కోర్ లోపల అదనపు పవర్ స్ట్రిప్‌లు పంపిణీ చేయబడతాయి [1].

ఈ దశలో పవర్ గ్రిడ్‌తో పాటు, కొన్ని భౌతిక-మాత్రమే సెల్‌లను చొప్పించబడతాయి. వెల్-ట్యాప్ సెల్‌లను కోర్ అంతటా ఉంచుతారు, తద్వారా సబ్‌స్ట్రేట్ మరియు బావులను VDD లేదా VSS కి కట్టివేస్తారు, ఇది లాచ్-అప్‌ను నివారిస్తుంది. కొనసాగింపును నిర్వహించడానికి మరియు సరిహద్దుల వద్ద డిజైన్ నియమ ఉల్లంఘనలను నివారించడానికి ప్లేస్‌మెంట్ వరుసల చివర్లలో ఎండ్-క్యాప్ సెల్‌లను ఉంచుతారు. ఇవి కలిసి విద్యుత్ స్థిరత్వాన్ని నిర్ధారిస్తాయి మరియు ప్రామాణిక-సెల్ ప్లేస్‌మెంట్ కోసం డిజైన్‌ను సిద్ధం చేస్తాయి.


Net
Snap
A
All Colors
V S
Favorite
Violation
Instance
Type
Block
StdCell
Cover
Physical
IO
Area IO
Black Box
Function
Status
Module
Cell
Blockage
Row
Floorplan
Partition
Power
Overlay
Track
Net
Route
Layer
Adaptive
Detail
Speed

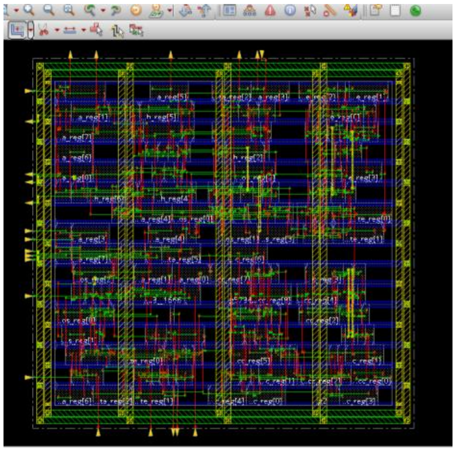
చిత్రం 3: UART డిజైన్ యొక్క ఫ్లోర్ ప్లానింగ్ మరియు పవర్ ప్లానింగ్

5. ప్లేస్‌మెంట్

ఫ్లోర్‌ప్లానింగ్ మరియు పవర్ ప్లానింగ్ తర్వాత, ప్లేస్‌మెంట్ దశ ప్రారంభమవుతుంది. ఈ దశలో, సింథసైజ్ చేయబడిన నెట్‌లిస్ట్ నుండి అన్ని ప్రామాణిక సెల్‌లకు కోర్ యొక్క ప్లేస్‌మెంట్ వరుసలలో భౌతిక స్థానాలు కేటాయించబడతాయి. ప్లేస్‌మెంట్ సాధనం వైర్‌లెంగ్త్, రద్దీ మరియు టైమింగ్ కోసం ఆప్టిమైజ్ చేస్తూ సెల్‌లను అమర్చుతుంది. చట్టబద్ధత ద్వారా ఓవర్‌లాప్‌లు పరిష్కరించబడతాయి మరియు టైమింగ్ పరిమితులను తీర్చడానికి క్లిష్టమైన మార్గాలు ఆప్టిమైజ్ చేయబడతాయి.

ఈ దశలో, టై-హై మరియు టై-లో సెల్స్ వంటి అదనపు భౌతిక-మాత్రమే సెల్స్ చొప్పించబడతాయి [2]. ఈ సెల్స్ పవర్ రైల్స్‌కు గేట్లను నేరుగా కనెక్ట్ చేయకుండా సురక్షితమైన స్థిరమైన లాజిక్ స్థాయిలను ('1' లేదా '0') అందిస్తాయి, తద్వారా విశ్వసనీయతను నిర్ధారిస్తాయి. కొన్ని ప్రవాహాలలో, భవిష్యత్ డిజైన్ సవరణలకు (ECO లు) మద్దతు ఇవ్వడానికి ప్లేస్‌మెంట్ సమయంలో స్పేర్ సెల్స్ కూడా జోడించబడవచ్చు.

ప్లేస్‌మెంట్ ఫలితంగా చట్టబద్ధమైన, సమయ-అవేర్ సెల్ అమరిక పవర్ రైల్స్‌తో సమలేఖనం చేయబడి, క్లాక్ ట్రీ సంశ్లేషణకు సిద్ధంగా ఉంటుంది.

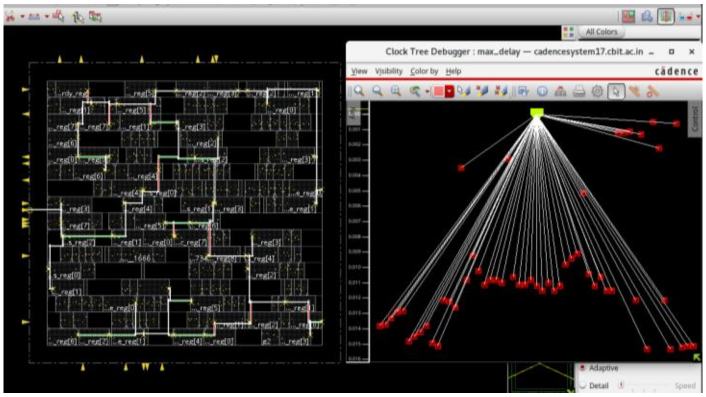


చిత్రం 4: నిర్వచించిన ఫ్లోర్‌ప్లాన్‌లో ప్రామాణిక సెల్‌ల స్థానం

6. క్లాక్ ట్రీ సింథసిస్ (CTS)

క్లాక్ ట్రీ సింథసిస్ (CTS) అనేది ప్లేస్‌మెంట్ తర్వాత క్లాక్ సిగ్నల్‌ను ఫ్లిప్-ఫ్లాప్‌లు మరియు రిజిస్టర్‌ల వంటి అన్ని వరుస అంశాలకు సమానంగా పంపిణీ చేయడానికి నిర్వహిస్తారు. క్లాక్ నెట్‌వర్క్ డిజైన్‌లో అత్యంత కీలకమైన భాగాలలో ఒకటి, ఎందుకంటే ఇది చిప్ అంతటా సమకాలీకరణను నియంత్రిస్తుంది. CTS సమయంలో, క్లాక్ సిగ్నల్‌ను వక్రీకరణ లేకుండా నడపడానికి ప్రత్యేక క్లాక్ బఫర్‌లు మరియు ఇన్వర్టర్‌లు సాధనం ద్వారా స్వయంచాలకంగా చొప్పించబడతాయి. CTS యొక్క ప్రాథమిక లక్ష్యాలు క్లాక్ స్కే [1] (వివిధ ఫ్లిప్-ఫ్లాప్‌ల వద్ద క్లాక్ సిగ్నల్ రాక సమయంలో వ్యత్యాసం) మరియు క్లాక్ లేటెన్సీ (క్లాక్ సోర్స్ నుండి సింక్‌లకు ఆలస్యం) తగ్గించడం.

UART యొక్క డిజైన్ ఫ్లోలో, CTS ఒక సమతుల్య క్రమానుగత ట్రీని ఉత్పత్తి చేస్తుంది, ఇక్కడ రూట్ క్లాక్ పిన్ అనేక బఫర్‌లను డ్రైవ్ చేస్తుంది, అవి అన్ని రిజిస్టర్‌లు క్లాక్‌ను స్వీకరించే వరకు సబ్-ట్రీలలోకి ఫ్యాన్ అవుట్ అవుతాయి. చొప్పించిన బఫర్‌లు లోడ్‌ను నియంత్రించడానికి, వక్రతను తగ్గించడానికి మరియు సమయ పరిమితులను తీర్చడానికి సహాయపడతాయి [5]. CTS తర్వాత, డిజైన్ సెటప్ మరియు హోల్డ్ అవసరాలు సంతృప్తి చెందాయని నిర్ధారించుకోవడానికి ఆప్టిమైజేషన్‌కు లోనవుతుంది, రూటింగ్ దశకు డిజైన్‌ను సిద్ధం చేస్తుంది.


All Colors
Clock Tree Debugger : max_delay — cadencesystem17.cbit.ac.in
View Visibility Color by Help
cadence
Adaptive
Detail
Speed

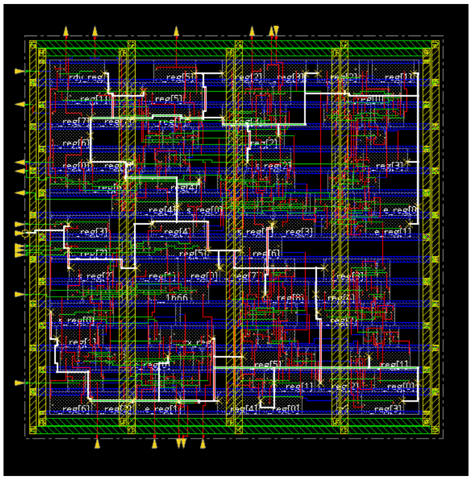
చిత్రం 5: UART డిజైన్ యొక్క క్లాక్ ట్రీ సింథసిస్ (CTS).

7. రూటింగ్

క్లాక్ ట్రీ సింథసిస్ తర్వాత, తదుపరి దశ రూటింగ్, ఇక్కడ నెట్‌లిస్ట్‌లో నిర్వచించబడిన అన్ని లాజికల్ కనెక్షన్‌లు చిప్‌లోని భౌతిక వైరింగ్‌లోకి అనువదించబడతాయి. రూటింగ్ రెండు దశల్లో నిర్వహించబడుతుంది: గ్లోబల్ రూటింగ్ మరియు వివరణాత్మక రూటింగ్[7].

- **గ్లోబల్ రూటింగ్‌లో** , రద్దీని నివారించి, వైర్‌లెంగ్త్‌ను బ్యాలెన్స్ చేస్తూ, ప్రతి నెట్‌కు సుమారుగా రూటింగ్ మార్గాలను సాధనం నిర్ణయిస్తుంది.
- **వివరణాత్మక రూటింగ్‌లో** , సాధనం నిర్దిష్ట లోహ పొరలపై ఖచ్చితమైన ట్రాక్‌లను కేటాయిస్తుంది మరియు వివిధ పొరలను కనెక్ట్ చేయడానికి వయాలను ఉంచుతుంది, డిజైన్ రూల్ చెక్‌లను (DRC) ఖచ్చితంగా అనుసరిస్తుంది.

రూటింగ్ సమయంలో, రెసిస్టెన్స్ మరియు కెపాసిటెన్స్ వంటి పరాన్నజీవులు తగ్గించబడతాయని కూడా సాధనం నిర్ధారించుకోవాలి, ఎందుకంటే అవి నేరుగా టైమింగ్‌ను ప్రభావితం చేస్తాయి. fig. 6 రూటెడ్ డిజైన్‌ను చూపుతుంది, ఇక్కడ సిగ్నల్ పాత్‌లు, క్లాక్ కనెక్షన్‌లు మరియు I/O ఇంటర్‌ఫేస్‌లు బహుళ లేయర్‌ల ఇంటర్‌కనెక్ట్ ఉపయోగించి కనెక్ట్ చేయబడతాయి. రూటింగ్ తర్వాత, తుది ధృవీకరణ మరియు GDSII జనరేషన్‌కు వెళ్లే ముందు అన్ని టైమింగ్, పవర్ మరియు సిగ్నల్ సమగ్రత అవసరాలు సంతృప్తి చెందాయని ధృవీకరించడానికి డిజైన్ వెలికితీత మరియు టైమింగ్ విశ్లేషణకు లోనవుతుంది.



చిత్రం 6: UART

డిజైన్ యొక్క రూటింగ్.

CC BY

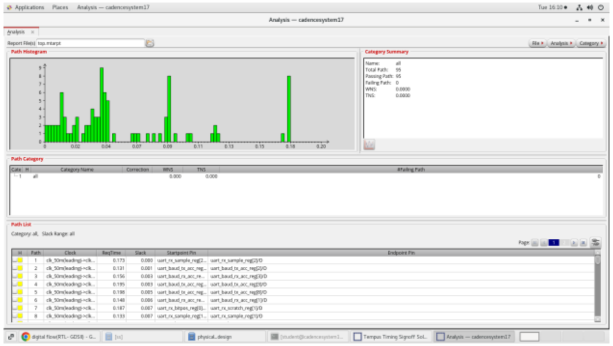
8. స్టాటిక్ టైమింగ్ అనాలిసిస్ (STA)

రూటింగ్ తర్వాత, తదుపరి దశ స్టాటిక్ టైమింగ్ అనాలిసిస్ (STA) టెంపస్ ఇంజిన్‌ను ఉపయోగించి సెటప్ మరియు హోల్డ్ అడ్డంకులను ధృవీకరించడానికి నిర్వహించబడుతుంది, ఇది టైమింగ్ క్లోజర్‌ను నిర్ధారిస్తుంది [1][6]. STA

కి పరీక్ష వెక్టర్‌లు అవసరం లేదు; బదులుగా, ఇది డిజైన్ అడ్డంకి ఫైల్ (SDC)లో అందించబడిన క్లాక్ నిర్వచనాలు మరియు సమయ అడ్డంకుల ఆధారంగా సర్క్యూట్‌లోని అన్ని సాధ్యమైన మార్గాలను విశ్లేషిస్తుంది.

ఈ సాధనం ప్రతి మార్గానికి రాక సమయం, అవసరమైన సమయం మరియు స్లాక్ వంటి ముఖ్యమైన సమయ పారామితులను లెక్కిస్తుంది. పాజిటివ్ స్లాక్ ఒక మార్గం సమయానికి అనుగుణంగా ఉందని సూచిస్తుంది, అయితే నెగటివ్ స్లాక్ ఉల్లంఘనను సూచిస్తుంది. చెత్త నెగటివ్ స్లాక్ (WNS) అత్యంత క్లిష్టమైన వైఫల్య మార్గాన్ని సూచిస్తుంది మరియు టోటల్ నెగటివ్ స్లాక్ (TNS) అన్ని ఉల్లంఘనల మొత్తాన్ని లెక్కిస్తుంది.

UART డిజైన్‌లో, STA ఫలితాలు విశ్లేషించబడిన 95 మార్గాలు ప్రయాణిస్తున్నాయని నిర్ధారిస్తాయి, WNS = 0 మరియు TNS = 0, పూర్తి సమయ ముగింపును సూచిస్తాయి. పాత్ హిస్టోగ్రాం వివిధ మార్గాల్లో జాప్యాల పంపిణీని వివరిస్తుంది మరియు పాత్ నివేదిక ప్రతి సమయ మార్గానికి ప్రారంభ బిందువులు, ముగింపు బిందువులు మరియు స్లాక్ విలువల గురించి వివరణాత్మక సమాచారాన్ని అందిస్తుంది. సున్నా సమయ ఉల్లంఘనలను సాధించడం వలన డిజైన్ తుది ధృవీకరణ మరియు GDSII జనరేషన్‌కు వెళ్లే ముందు లక్ష్య గడియార పౌనఃపున్యంలో విశ్వసనీయంగా పనిచేయగలదని ధృవీకరిస్తుంది.


Applications
Places
Analysis — cadencesystem17
Tue 16:10
Analysis — cadencesystem17
Analysis
File
Analysis
Category
Path Histogram
Category Summary
Path Category
Path List
Page: 1
digital flow(RTL- GDSII) - G...
[ss]
physical_design
[student@cadencesystem1...
Tempus Timing Signoff Sol...
Analysis — cadencesystem17

Report File(s): top.mtarpt

Name: all
Total Path: 95
Passing Path: 95
Failing Path: 0
WNS: 0.0000
TNS: 0.0000

Cate	H	Category Name	Correction	WNS	TNS	#Failing Path
1		all		0.000	0.000	0

Category: all, Slack Range: all

H	Path	Clock	ReqTime	Slack	Startpoint Pin	Endpoint Pin
🟨	1	clk_50m(leading)->clk...	0.173	0.000	uart_rx_sample_reg[2...	uart_rx_sample_reg[2]/D
🟨	2	clk_50m(leading)->clk...	0.131	0.001	uart_baud_tx_acc_reg...	uart_baud_tx_acc_reg[2]/D
🟨	3	clk_50m(leading)->clk...	0.156	0.003	uart_baud_rx_acc_re...	uart_baud_rx_acc_reg[3]/D
🟨	4	clk_50m(leading)->clk...	0.195	0.003	uart_baud_tx_acc_reg...	uart_baud_tx_acc_reg[3]/D
🟨	5	clk_50m(leading)->clk...	0.198	0.005	uart_baud_tx_acc_reg...	uart_baud_tx_acc_reg[8]/D
🟨	6	clk_50m(leading)->clk...	0.148	0.006	uart_baud_rx_acc_re...	uart_baud_rx_acc_reg[1]/D
🟨	7	clk_50m(leading)->clk...	0.187	0.007	uart_rx_bitpos_reg[0]...	uart_rx_scratch_reg[1]/D
🟨	8	clk_50m(leading)->clk...	0.133	0.007	uart_rx_sample_reg[1...	uart_rx_sample_reg[1]/D

Fig.7: కాడెన్స్ టెంపస్ ఉపయోగించి UART డిజైన్ యొక్క స్టాటిక్ టైమింగ్ అనాలిసిస్ (STA).

9. భౌతిక ధృవీకరణ

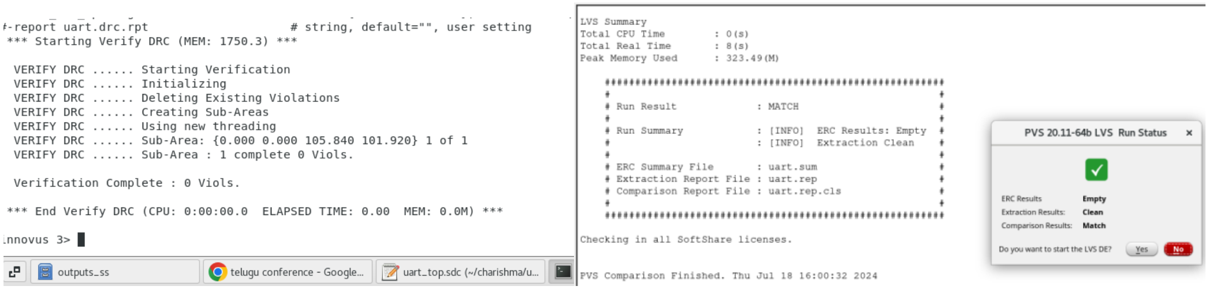
భౌతిక ధృవీకరణ అనేది అమలు ప్రవాహం యొక్క చివరి దశ, డిజైన్ తయారీ నియమాలకు అనుగుణంగా ఉందని మరియు దాని స్కీమాటిక్ వివరణకు తార్కికంగా సమానంగా ఉందని నిర్ధారిస్తుంది. కాడెన్స్ ఇన్నోవస్‌లో ప్లేస్‌మెంట్, రూటింగ్ మరియు టైమింగ్ క్లోజర్‌ను పూర్తి చేసిన తర్వాత, తుది లేఅవుట్ డేటాబేస్ సైన్‌ఆఫ్ ధృవీకరణ కోసం కాడెన్స్ పర్చుసోలోకి ఎగుమతి చేయబడుతుంది మరియు దిగుమతి చేయబడుతుంది.

రెండు ప్రాథమిక తనిఖీలు నిర్వహించబడతాయి:

- **డిజైన్ రూల్ చెక్ (DRC):** లేఅవుట్ కనీస అంతరం, వెడల్పు మరియు ఎన్‌క్లోజర్ నియమాలు వంటి ఫౌండ్రీ ప్రక్రియ యొక్క రేఖాగణిత పరిమితులకు కట్టుబడి ఉందని ధృవీకరిస్తుంది. లితోగ్రఫీ లోపాలు లేకుండా లేఅవుట్ తయారు చేయదగినదని DRC నిర్ధారిస్తుంది [1].

- **లేఅవుట్ వర్సెస్ స్కీమాటిక్ (LVS):** భౌతిక లేఅవుట్ సంశ్లేషణ తర్వాత ఉత్పత్తి చేయబడిన లాజికల్ నెట్‌లిస్ట్‌తో సరిపోలుతుందని నిర్ధారిస్తుంది. LVS పరికర కనెక్టివిటీ, పిన్ కరస్పాండెన్స్ మరియు లేఅవుట్ మరియు నెట్‌లిస్ట్ మధ్య సోపానక్రమ స్థిరత్వాన్ని ధృవీకరిస్తుంది [1].

మెంటర్ కాలిబర్‌ను ఉపయోగించి నిర్వహించబడ్డాయి . నివేదికలు సున్నా DRC లోపాలు మరియు LVS సరిపోలిక-రహిత ఫలితాలతో క్లీన్ డిజైన్‌ను నిర్ధారించాయి. అందువలన, లేఅవుట్ పూర్తిగా ధృవీకరించబడింది మరియు GDSII జనరేషన్ మరియు టేప్-అవుట్ కోసం సిద్ధంగా ఉంది.



```
#-report uart.drc.rpt                    # string, default="", user setting
 *** Starting Verify DRC (MEM: 1750.3) ***

  VERIFY DRC ...... Starting Verification
  VERIFY DRC ...... Initializing
  VERIFY DRC ...... Deleting Existing Violations
  VERIFY DRC ...... Creating Sub-Areas
  VERIFY DRC ...... Using new threading
  VERIFY DRC ...... Sub-Area: {0.000 0.000 105.840 101.920} 1 of 1
  VERIFY DRC ...... Sub-Area : 1 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:00.0  ELAPSED TIME: 0.00  MEM: 0.0M) ***

innovus 3>
```

```
LVS Summary
Total CPU Time          : 0(s)
Total Real Time         : 8(s)
Peak Memory Used        : 323.49(M)

    #######################################################
    #                                                     #
    # Run Result               : MATCH                    #
    #                                                     #
    # Run Summary              : [INFO]  ERC Results: Empty #
    #                          : [INFO]  Extraction Clean #
    #                                                     #
    # ERC Summary File         : uart.sum                 #
    # Extraction Report File : uart.rep                   #
    # Comparison Report File : uart.rep.cls               #
    #                                                     #
    #######################################################

Checking in all SoftShare licenses.

PVS Comparison Finished. Thu Jul 18 16:00:32 2024
```

PVS 20.11-64b LVS Run Status

ERC Results: Empty
Extraction Results: Clean
Comparison Results: Match

Do you want to start the LVS DE?

Fig.8: Fig.9 లో DRC (డిజైన్ రూల్ చెక్) ధృవీకరణ: కాడెన్స్ ఇన్నోవస్‌లో LVS (లేఅవుట్ vs . స్కీమాటిక్) ధృవీకరణ కాడెన్స్ ఇన్నోవస్

10. GDSII జనరేషన్

చివరి దశలో చిప్ ఫ్యాబ్రికేషన్ కోసం పరిశ్రమ-ప్రామాణిక ఫార్మాట్ అయిన GDSII ఫైల్ ఉత్పత్తి అవుతుంది. ఈ ఫైల్ లేయర్‌లు, బహుభుజాలు, వియాస్ మరియు ఇంటర్‌కనెక్ట్‌లతో సహా పూర్తి లేఅవుట్‌ను సూచిస్తుంది. ఇన్‌పుట్‌లు ధృవీకరించబడిన లేఅవుట్ డేటాబేస్, మరియు అవుట్‌పుట్ టేప్-అవుట్ సిద్ధంగా ఉన్న GDSII ఫైల్, ఇది ఫ్యాబ్రికేషన్ కోసం సెమీకండక్టర్ ఫౌండ్రీకి పంపబడుతుంది .

IV ముగింపు

ఈ పనిలో, కాడెన్స్ సాధనాలను ఉపయోగించి పూర్తి RTL-to-GDSII డిజిటల్ డిజైన్ ప్రవాహాన్ని విజయవంతంగా అమలు చేశారు. RTL డిజైన్ మరియు సంశ్లేషణ నుండి ప్రారంభించి, ఫ్లోర్‌ప్లానింగ్, పవర్ ప్లానింగ్, ప్లేస్‌మెంట్, క్లాక్ ట్రీ సంశ్లేషణ మరియు రూటింగ్ ద్వారా డిజైన్ క్రమంగా మెరుగుపరచబడింది. స్టాటిక్ టైమింగ్ విశ్లేషణ ద్వారా టైమింగ్ క్లోజర్ సాధించబడింది, అన్ని సమయ పరిమితులు నెరవేరాయని నిర్ధారిస్తుంది. చివరగా, DRC మరియు LVS తనిఖీలతో సహా భౌతిక ధృవీకరణ నిర్వహించబడింది, రెండూ ఉల్లంఘనలు లేకుండా ఆమోదించబడ్డాయి, డిజైన్ యొక్క ఖచ్చితత్వం మరియు తయారీ సామర్థ్యాన్ని నిర్ధారిస్తాయి.

ప్రస్తావనలు

[1] ఎన్. వెస్టే మరియు డి. హారిస్, *CMOS VLSI డిజైన్: ఎ సర్క్యూట్స్ అండ్ సిస్టమ్స్ పెర్స్పెక్టివ్* , 4వ ఎడిషన్. బోస్టన్, MA : అడిసన్-వెస్లీ, 2011.
[2] జె.ఎం. రాబే, ఎ. చంద్రకాసన్, మరియు బి. నికోలిక్, *డిజిటల్ ఇంటిగ్రేటెడ్ సర్క్యూట్స్: ఎ డిజైన్ పెర్స్పెక్టివ్* , 2వ ఎడిషన్. అప్పర్ సాడిల్ రివర్, NJ : ప్రెంటిస్ హాల్, 2003.
[3] కె. ఉమాపతి మరియు పి. లక్ష్మి, "వెరిలాగ్ ఉపయోగించి UART రూపకల్పన మరియు అమలు," *ఇంటర్నేషనల్ జర్నల్ ఆఫ్ ఇంజనీరింగ్ అండ్ టెక్నాలజీ (IJET)* , వాల్యూమ్. 5, నం. 3, పేజీలు. 2321–2326, 2013.
[4] ఎస్. పల్నిట్కర్, *వెరిలాగ్ HDL : ఎ గైడ్ టు డిజిటల్ డిజైన్ అండ్ సింథసిస్* , 2వ ఎడిషన్. అప్పర్ సాడిల్ రివర్, NJ : ప్రెంటిస్ హాల్, 2003.
[5] కాడెన్స్ డిజైన్ సిస్టమ్స్, *కాడెన్స్ ఇన్నోవస్ ఇంప్లిమెంటేషన్ సిస్టమ్ యూజర్ గైడ్* , కాడెన్స్, 2023.
[6] కాడెన్స్ డిజైన్ సిస్టమ్స్, *కాడెన్స్ జెనస్ సింథసిస్ సొల్యూషన్ యూజర్ గైడ్* , కాడెన్స్, 2023.
[7] సిమెన్స్ EDA, *కాలిబర్ ఫిజికల్ వెరిఫికేషన్ యూజర్ మాన్యువల్* , సిమెన్స్, 2023.

CC BY

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025
PP 166-174

RTL-to-GDSII Implementation of a Universal Asynchronous Receiver-Transmitter (UART) in SCL

Pandala Charishma, Dr. Mohd Ziauddin Jahangir, Dr. D Krishna Reddy
VLSID, Dept of ECE, Chaitanya Bharathi Institute of Technology,Hyderabad, Telangana-India
Corresponding Author : charishmapandala2025@gmail.com

Abstract

This paper presents the complete digital flow implementation of a Universal Asynchronous Receiver-Transmitter (UART) from RTL to GDS II using SCL PDK. UART is one of the most widely used serial communication protocols in embedded and VLSI systems. The proposed UART design consists of a transmitter, receiver, and baud rate generator. The design was modeled in Verilog, functionally verified through simulation, synthesized, and implemented using the complete RTL-to-GDSII digital flow in Cadence tools. Post-layout results confirm successful timing closure and generation of a manufacturable GDSII, demonstrating the suitability of the design for ASIC and FPGA applications.The complete flow is implemented using CADENCE tools and physical verification is performed using siemens CALIBRE.

Keywords

UART, Digital Design Flow, RTL-to-GDSII, Verilog HDL, ASIC, FPGA

I. Introduction

Serial communication is essential in modern digital systems, enabling data transfer between processors, peripherals, and external devices [3]. Among the available protocols, UART is widely adopted due to its simplicity, low pin count, and ease of implementation [4].

This paper demonstrates the complete digital design flow (RTL to GDSII) using a UART as a case study. By applying industry-standard CAD tools, the design is carried through all stages, including RTL modeling, functional simulation, logic synthesis, floorplanning, placement and routing, static timing analysis (STA), and GDSII generation.

II. Background and Related Work

UART communication consists of transmitting and receiving serial data with start, data, parity (optional), and stop bits. Our work emphasizes the complete RTL-to-GDSII implementation, highlighting how a commonly used communication block can be taken from behavioral description to layout.

III. Methodology

The development of the UART design was carried out using a standard RTL-to-GDSII digital design flow. Each step in the methodology progressively refines the hardware description from an abstract behavioral model to a manufacturable layout. The flow can be divided into the following major stages:

1. Register Transfer Level (RTL) Design

At the initial stage, the UART is described in a hardware description language (HDL), specifically Verilog. The RTL model captures the functional behavior and timing of the UART, including modules such as transmitter, receiver and baud rate generator.

The RTL serves as the golden reference design. Inputs at this stage include the Verilog source code, a project configuration file, and a simulation testbench for functional verification. The output is a verified RTL model that can be synthesized into a gate-level description.

2. RTL Simulation and Verification

The first stage after writing the Verilog description of the UART is functional simulation. This step ensures that the design behavior matches the intended UART protocol specification before moving to synthesis. A testbench was developed to apply stimulus to the design by enabling transmission and providing data words on the input bus (data[7:0]). The simulation also monitored the output bus (rxdata[7:0]), along with the control signals (rd y, rd y_clr, and tx_busy).

The simulation waveform, shown in Figure 1, demonstrates correct operation of the UART. Data words ranging from 00h to 0Fh are sequentially transmitted on the input bus. The receiver successfully reconstructs these words on the rxdata output, validating correct transmission and reception. The tx_busy signal remains high during active transmission, and the rd y flag asserts upon successful data reception, indicating proper synchronization between transmitter and receiver modules.

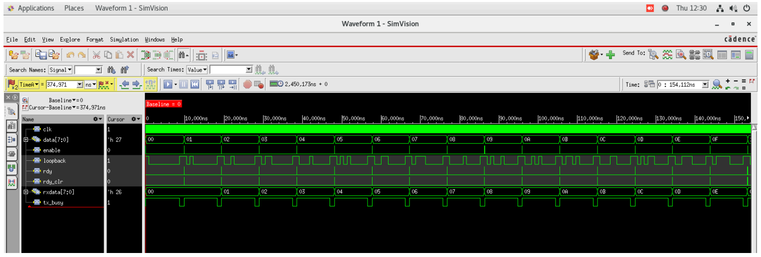

Applications
Places
Waveform 1 - SimVision
Thu 12:30
Waveform 1 - SimVision
File Edit View Explore Format Simulation Windows Help
cādence
Send To:
Search Names: Signal
Search Times: Value
TimeA = 374,971 ns
2,450,173ns + 0
Time: 0 : 154,112ns
Baseline = 0
Cursor-Baseline = 374,971ns
Name
Cursor
clk 1
data[7:0] 'h 27
enable 0
loopback 1
rdy 0
rdy_clr 0
rxdata[7:0] 'h 26
tx_busy 1
Baseline = 0
0
10,000ns
20,000ns
30,000ns
40,000ns
50,000ns
60,000ns
70,000ns
80,000ns
90,000ns
100,000ns
110,000ns
120,000ns
130,000ns
140,000ns
150,
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E

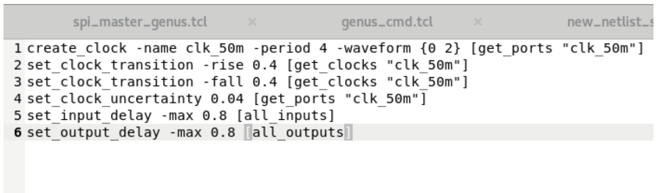
Fig.1:simulation of UART in incisive cadence tool

This stage produces a functionally verified RTL model, which serves as the golden design for downstream implementation steps. Only after verifying this correctness can the design be synthesized into gate-level netlists.

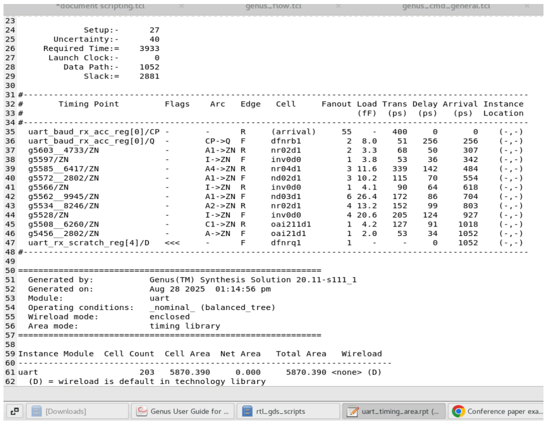
3. Logic Synthesis

Once the functional correctness of the RTL design was established through simulation, the UART description was synthesized into a gate-level netlist using the Cadence Genus tool. Synthesis transforms the behavioral Verilog into a technology-dependent description mapped onto the standard cell library of the chosen technology node.

The inputs to synthesis included the UART RTL, the technology library in Liberty (.lib) format, and a set of timing and clock constraints defined in Synopsys Design Constraints (SDC) format. These constraints specify the operating frequency, input/output delays, and design requirements.



```
create_clock -name clk_50m -period 4 -waveform {0 2} [get_ports "clk_50m"]
set_clock_transition -rise 0.4 [get_clocks "clk_50m"]
set_clock_transition -fall 0.4 [get_clocks "clk_50m"]
set_clock_uncertainty 0.04 [get_ports "clk_50m"]
set_input_delay -max 0.8 [all_inputs]
set_output_delay -max 0.8 [all_outputs]
```



```

            Setup:-      27
      Uncertainty:-      40
    Required Time:=    3933
     Launch Clock:-       0
        Data Path:-    1052
            Slack:=    2881

#---------------------------------------------------------------------------------------------------------
#       Timing Point          Flags    Arc    Edge   Cell       Fanout Load Trans Delay Arrival Instance
#                                                                      (fF)  (ps)  (ps)   (ps)  Location
#---------------------------------------------------------------------------------------------------------
  uart_baud_rx_acc_reg[0]/CP -        -       R     (arrival)      55    -   400     0       0    (-,-)
  uart_baud_rx_acc_reg[0]/Q  -        CP->Q   F     dfnrb1          2  8.0    51   256     256    (-,-)
  g5603__4733/ZN             -        A1->ZN R     nr02d1          2  3.3    68    50     307    (-,-)
  g5597/ZN                   -        I->ZN  F     inv0d0          1  3.8    53    36     342    (-,-)
  g5585__6417/ZN             -        A4->ZN R     nr04d1          3 11.6   339   142     484    (-,-)
  g5572__2802/ZN             -        A1->ZN F     nd02d1          3 10.2   115    70     554    (-,-)
  g5566/ZN                   -        I->ZN  R     inv0d0          1  4.1    90    64     618    (-,-)
  g5562__9945/ZN             -        A1->ZN F     nd03d1          6 26.4   172    86     704    (-,-)
  g5534__8246/ZN             -        A2->ZN R     nr02d1          4 13.2   152    99     803    (-,-)
  g5528/ZN                   -        I->ZN  F     inv0d0          4 20.6   205   124     927    (-,-)
  g5508__6260/ZN             -        C1->ZN R     oai211d1        1  4.2   127    91    1018    (-,-)
  g5456__2802/ZN             -        A->ZN  F     oai21d1         1  2.0    53    34    1052    (-,-)
  uart_rx_scratch_reg[4]/D   <<<      -       F     dfnrq1          1    -     -     0    1052    (-,-)
#---------------------------------------------------------------------------------------------------------

============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Aug 28 2025  01:14:56 pm
  Module:                 uart
  Operating conditions:   _nominal_ (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
--------------------------------------------------------------------------
uart                    203   5870.390     0.000     5870.390 <none> (D)
  (D) = wireload is default in technology library
```

Fig.2: Synopsis design constrains file

Fig.3: synthesis report in genus tool

The output of this stage is a gate-level netlist expressed entirely in terms of standard cells such as flip-flops, NAND, NOR, and multiplexers.netlist will be in the form of <ddesign>_netlist.v. In addition, detailed timing, area, and power reports are generated to assess whether the design meets its performance goals.

The synthesis report, shown in Figure 3, presents the critical timing path of the UART. The datapath delay is reported as 1052 ps, while the required time is 3933 ps, yielding a positive slack of 2881 ps. This indicates that the design comfortably meets the target timing specifications without violations. The report also summarizes the cell count and area utilization, with the synthesized UART comprising 203 standard cells and occupying an area of 5870.39 µm².

Thus, at the end of this stage, the UART is available as a gate-level netlist that satisfies timing and area constraints, and this netlist serves as the input to physical design.

4. Floorplanning and Power Planning

Floorplanning defines the initial physical layout of the chip. The die size is estimated based on utilization reported during synthesis, and the UART core region is defined accordingly. Input/output pins are placed around the periphery to satisfy interface requirements. At this stage, power planning is also performed to create a robust power distribution network. Power rings for VDD and VSS are constructed around the core boundary, horizontal and vertical power rails are inserted across placement rows, and additional power stripes are distributed within the core to ensure uniform voltage delivery[1].

In addition to the power grid, certain physical-only cells are inserted during this stage. Well-tap cells are placed across the core to tie the substrate and wells to VDD or VSS, preventing latch-up. End-cap cells are positioned at the ends of placement rows to maintain continuity and avoid design rule violations at boundaries. Together, these ensure electrical stability and prepare the design for standard-cell placement.

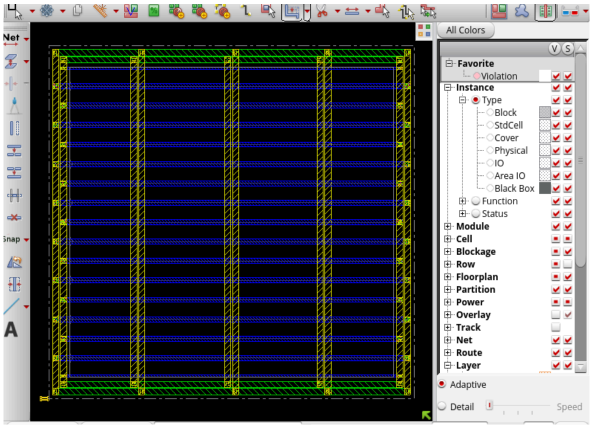

Net
Snap
All Colors
V S
Favorite
Violation
Instance
Type
Block
StdCell
Cover
Physical
IO
Area IO
Black Box
Function
Status
Module
Cell
Blockage
Row
Floorplan
Partition
Power
Overlay
Track
Net
Route
Layer
Adaptive
Detail
Speed

Fig.3 :Floorplanning and power planning of UART design

5. Placement

After floorplanning and power planning, the placement stage begins. In this step, all standard cells from the synthesized netlist are assigned physical locations within the placement rows of the core. The placement tool arranges cells while optimizing for wirelength, congestion, and timing. Overlaps are resolved through legalization, and critical paths are optimized to meet timing constraints.

During this stage, additional physical-only cells such as tie-high and tie-low cells are inserted [2]. These cells provide safe constant logic levels ('1' or '0') without directly connecting gates to the power rails, thus ensuring reliability. In some flows, spare cells may also be added during placement to support future design modifications (ECOs).

The result of placement is a legalized, timing-aware cell arrangement aligned with the power rails, ready for clock tree synthesis.

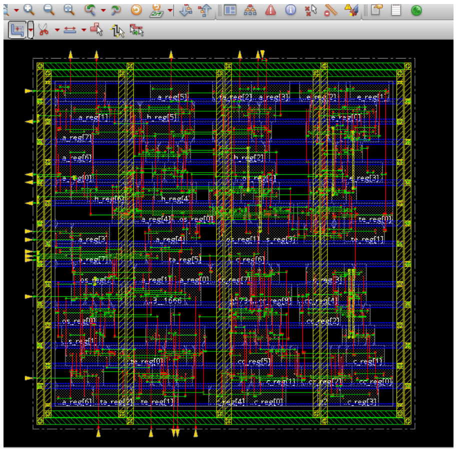


Fig. 4: Placement of standard cells within the defined floorplan

6. Clock Tree Synthesis (CTS)

Clock Tree Synthesis (CTS) is performed after placement to distribute the clock signal evenly to all sequential elements such as flip-flops and registers. The clock network is one of the most critical parts of the design, as it governs synchronization across the chip. During CTS, special clock buffers and inverters are inserted automatically by the tool to drive the clock signal without distortion. The primary objectives of CTS are to minimize clock skew [1] (the difference in arrival time of the clock signal at different flip-flops) and clock latency (the delay from the clock source to the sinks).

In the design flow of the UART, CTS generated a balanced hierarchical tree where the root clock pin drives several buffers that fan out into sub-trees until all registers receive the clock. The inserted buffers help control load, reduce skew, and meet timing constraints [5]. After CTS, the design undergoes optimization to ensure that setup and hold requirements are satisfied, preparing the design for the routing stage.

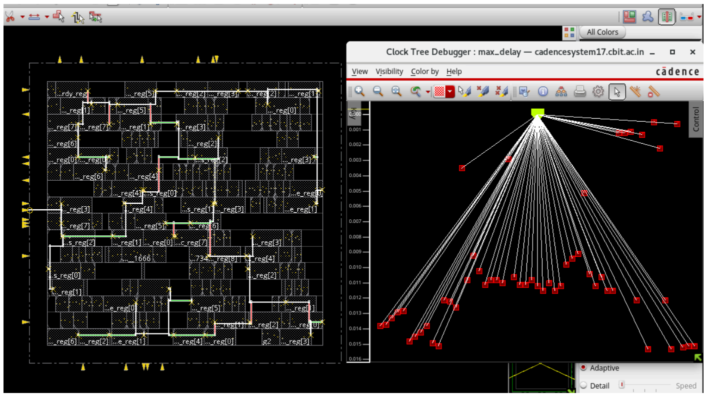


Fig.5 : Clock Tree Synthesis (CTS) of the UART design

7. Routing

After Clock Tree Synthesis, the next step is Routing, where all logical connections defined in the netlist are translated into physical wiring on the chip. Routing is carried out in two stages: global routing and detailed routing[7].

- In **global routing**, the tool determines approximate routing paths for each net while avoiding congestion and balancing wirelength.
- In **detailed routing**, the tool assigns exact tracks on specific metal layers and places vias to connect different layers, strictly following Design Rule Checks (DRC).

During routing, the tool must also ensure that parasitics such as resistance and capacitance are minimized, as they directly impact timing. The fig.6 shows the routed design, where signal paths, clock connections, and I/O interfaces are connected using multiple layers of interconnect. Post-routing, the design undergoes extraction and timing analysis to verify that all timing, power, and signal integrity requirements are satisfied before proceeding to final verification and GDSII generation.

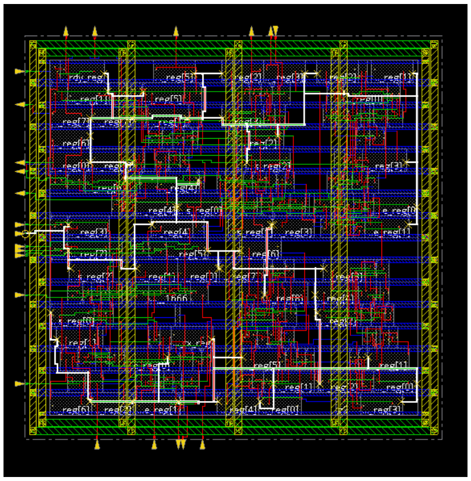


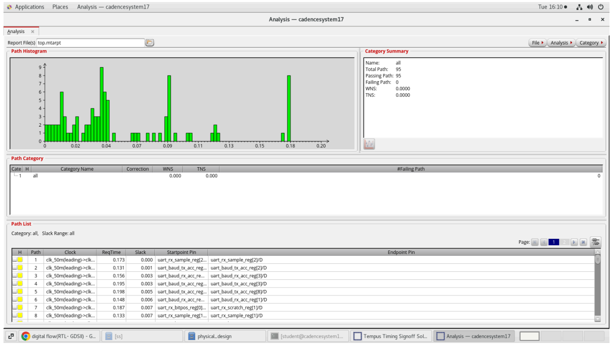
Fig. 6: Routing of the UART design.

8. Static Timing Analysis (STA)

After routing, the next step is Static Timing Analysis (STA) is performed to verify setup and hold constraints using the Tempus engine, confirming timing closure [1][6]. STA does not require test vectors; instead, it analyzes all possible paths in the circuit based on the clock definitions and timing constraints provided in the design constraint file (SDC).

The tool calculates important timing parameters such as arrival time, required time, and slack for each path. Positive slack indicates that a path meets timing, while negative slack indicates a violation. The Worst Negative Slack (WNS) represents the most critical failing path, and the Total Negative Slack (TNS) quantifies the sum of all violations.

In the UART design, STA results confirm that all 95 analyzed paths are passing, with WNS = 0 and TNS = 0, indicating complete timing closure. The path histogram illustrates the distribution of delays across different paths, and the path report provides detailed information about startpoints, endpoints, and slack values for each timing path. Achieving zero timing violations validates that the design can reliably operate at the target clock frequency before proceeding to final verification and GDSII generation.


Applications
Places
Analysis — cadencesystem17
Tue 16:10
Analysis — cadencesystem17
Analysis
File
Analysis
Category
Path Histogram
9
8
7
6
5
4
3
2
1
0
0
0.02
0.04
0.07
0.09
0.11
0.13
0.15
0.18
0.20
Category Summary
Path Category
Path List
Page: 1 2
digital flow(RTL- GDSII) - G...
[ss]
physical_design
[student@cadencesystem1...
Tempus Timing Signoff Sol...
Analysis — cadencesystem17

Report File(s) top.mtarpt

Name: all
Total Path: 95
Passing Path: 95
Failing Path: 0
WNS: 0.0000
TNS: 0.0000

Cate	H	Category Name	Correction	WNS	TNS	#Failing Path
1		all		0.000	0.000	0

Category: all, Slack Range: all

H	Path	Clock	ReqTime	Slack	Startpoint Pin	Endpoint Pin
	1	clk_50m(leading)->clk...	0.173	0.000	uart_rx_sample_reg[2...	uart_rx_sample_reg[2]/D
	2	clk_50m(leading)->clk...	0.131	0.001	uart_baud_tx_acc_reg...	uart_baud_tx_acc_reg[2]/D
	3	clk_50m(leading)->clk...	0.156	0.003	uart_baud_rx_acc_re...	uart_baud_rx_acc_reg[3]/D
	4	clk_50m(leading)->clk...	0.195	0.003	uart_baud_tx_acc_reg...	uart_baud_tx_acc_reg[3]/D
	5	clk_50m(leading)->clk...	0.198	0.005	uart_baud_tx_acc_reg...	uart_baud_tx_acc_reg[8]/D
	6	clk_50m(leading)->clk...	0.148	0.006	uart_baud_rx_acc_re...	uart_baud_rx_acc_reg[1]/D
	7	clk_50m(leading)->clk...	0.187	0.007	uart_rx_bitpos_reg[0]...	uart_rx_scratch_reg[1]/D
	8	clk_50m(leading)->clk...	0.133	0.007	uart_rx_sample_reg[1...	uart_rx_sample_reg[1]/D

Fig.7:Static Timing Analysis (STA) of the UART design using Cadence Tempus

9. Physical Verification

Physical verification is the final stage of the implementation flow, ensuring that the design complies with manufacturing rules and is logically equivalent to its schematic description. After completing placement, routing, and timing closure in Cadence Innovus, the final layout database is exported and imported into Cadence Virtuoso for signoff verification.

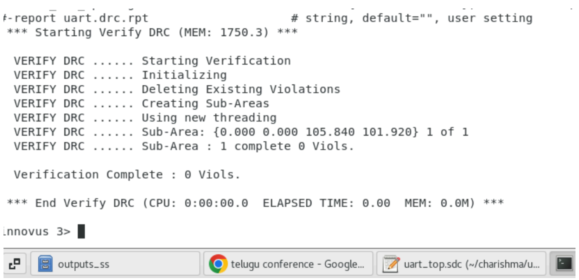
Two primary checks are carried out:

- **Design Rule Check (DRC):** Validates that the layout adheres to the geometric constraints of the foundry process, such as minimum spacing, width, and enclosure rules. DRC ensures that the layout is manufacturable without lithography errors [1].

- **Layout versus Schematic (LVS):** Confirms that the physical layout matches the logical netlist generated after synthesis. LVS verifies device connectivity, pin correspondence, and hierarchy consistency between layout and netlist [1].

These checks were performed using **Mentor Calibre**, a widely adopted physical verification tool. The reports confirmed a clean design, with zero DRC errors and LVS mismatch-free results. Thus, the layout was fully verified and ready for GDSII generation and tape-out.

CC BY



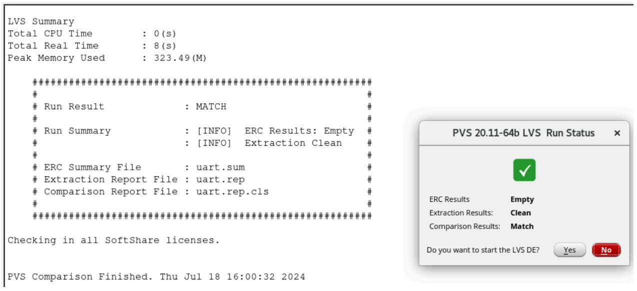
```
#-report uart.drc.rpt                    # string, default="", user setting
 *** Starting Verify DRC (MEM: 1750.3) ***

  VERIFY DRC ...... Starting Verification
  VERIFY DRC ...... Initializing
  VERIFY DRC ...... Deleting Existing Violations
  VERIFY DRC ...... Creating Sub-Areas
  VERIFY DRC ...... Using new threading
  VERIFY DRC ...... Sub-Area: {0.000 0.000 105.840 101.920} 1 of 1
  VERIFY DRC ...... Sub-Area : 1 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:00.0  ELAPSED TIME: 0.00  MEM: 0.0M) ***

innovus 3>
```



```
LVS Summary
Total CPU Time        : 0(s)
Total Real Time       : 8(s)
Peak Memory Used      : 323.49(M)

    #######################################################
    #                                                     #
    # Run Result             : MATCH                      #
    #                                                     #
    # Run Summary            : [INFO]  ERC Results: Empty #
    #                        : [INFO]  Extraction Clean   #
    #                                                     #
    # ERC Summary File       : uart.sum                   #
    # Extraction Report File : uart.rep                   #
    # Comparison Report File : uart.rep.cls               #
    #                                                     #
    #######################################################

Checking in all SoftShare licenses.


PVS Comparison Finished. Thu Jul 18 16:00:32 2024
```

PVS 20.11-64b LVS Run Status

ERC Results: Empty
Extraction Results: Clean
Comparison Results: Match

Do you want to start the LVS DE?

Fig.8: DRC (Design Rule Check) verification in Cadence Innovus

Fig.9 : LVS (Layout vs. Schematic) verification in Cadence Innovus

10. GDSII Generation

The final stage produces the GDSII file, which is the industry-standard format for chip fabrication. This file represents the complete layout, including layers, polygons, vias, and interconnects. Inputs are the verified layout database, and the output is the tape-out ready GDSII file, which is sent to the semiconductor foundry for fabrication.

IV Conclusion

In this work, a complete RTL-to-GDSII digital design flow was successfully implemented using Cadence tools. Starting from RTL design and synthesis, the design was progressively refined through floorplanning, power planning, placement, clock tree synthesis, and routing. Timing closure was achieved through static timing analysis, ensuring that all timing constraints were met. Finally, physical verification was carried out, including DRC and LVS checks, both of which passed without violations, confirming the correctness and manufacturability of the design.

References

[1] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Boston, MA: Addison-Wesley, 2011.
[2] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.
[3] K. Umapathy and P. Lakshmi, "Design and Implementation of UART Using Verilog," *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 3, pp. 2321–2326, 2013.
[4] S. Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.
[5] Cadence Design Systems, *Cadence Innovus Implementation System User Guide*, Cadence, 2023.
[6] Cadence Design Systems, *Cadence Genus Synthesis Solution User Guide*, Cadence, 2023.
[7] Siemens EDA, *Calibre Physical Verification User Manual*, Siemens, 2023.