Systems On-Chip

Lab 6
Top SoC design

Bologna Master Degree in Electrical and Computer Engineering (MEEC)
Instituto Superior Técnico
1st semester, 2nd period

Group 8
David Benjamin Krause | 104898
Duarte Miguel de Aguiar Pinto e Morais Marques | 96523
Eduardo Daniel Roque Martins | 96923

January 16th 2023

1 Introduction

The main objective of this work is to use Cadence Innovus and TCL scripts to automatically and properly generate the **battery charger core** and the **charger top**. For this purpose, simulations were initially run with the 4-bit counter, using the `.tar` files copied from the directory `/afs/ist.utl.pt/users/6/1/ist13261/public`. These files were then used as a starting point for the battery charger, having the LEF files of the different blocks (`BATCHARGERbg.lef`, `BATCHARGERctr.lef`, `BATCHARGERpower.lef` and `BATCHARGERsaradc.lef`) and the verilog of the charger top (`BATCHARGERcore.v`) been also obtained from this directory. It was also important to use the UMC 130nm documentation in the directory `/opt/ic_tools/pdk/faraday/umc130/HS`.

Throughout this report, the files `innovus_BATCHARGERcore` (TCL script) and `BATCHARGERcore.lef` (automatically generated LEF file) - regarding the charger core - are shown. Moreover, regarding the full chip, the files `BATCHARGER.save.io` (with the relative positions of the IO cells), `BATCHARGER.v` (with the designations of the IO cells), `innovus_BATCHARGER` (TCL script) and `BATCHARGER_init_-shield.rpt` (regarding the shielding process) are also present. All these files are included in the zip delivered in the submission for this laboratory assignment, for easier access, along with two other relevant (and lengthy) files - `innovus_core.log` (log file from the charger core creation in Innovus) and `innovus_BATCHARGER.log` (log file from the full chip creation in Innovus).

2 Charger core

The **charger core** routing and LEF file are automatically generated using the TCL script shown in Figure 1 (in which the unused `.tcl` termination is not necessary). Once inside the newly created directory `BATCHARGERcore_v_lefs`, a script is run with the command `source /opt/ic_-tools/init/init-innovus20-11-hf000`, after which the tool is initiated with the command `innovus`. With `source innovus_BATCHARGERcore`, the results shown in Figures 2 to 4 can then be obtained. Some of the commands used in the script were adapted from the 4-bit counter, while most others were obtained by using the GUI interface and the successively generated `.cmd` files.

Initially, the **floorplan** is created with a width of $640\mu\text{m}$ - a multiple of $0.4\mu\text{m}$, as indicated in the Innovus tutorial from the previous laboratory assignment - and a height of $470.4\mu\text{m}$ - a multiple of $3.2\mu\text{m}$. Additionally, core to IO margins of $5.6\mu\text{m}$ (multiple of $0.4\mu\text{m}$) are used. The width and height of the cell were defined taking into account the width and height of the different blocks, imposed in the respective LEF files (obtained from the public directory). The positioning of the blocks is shown in Figure 2, in which it can seen that the instance `BATCHbg` - a **sensitive area**, which generates the reference voltage, reference current, clock and power-on reset - was placed far away from `BATCHpower` - a **noisy area** (analog block that controls the charging current). All the blocks were also spaced away enough from each other so that the routing process could occur without issues; moreover, enough space is given to facilitate access for "special pins". For instance, the pin `iforcedbat` in `BATCHpower` (in which a considerably higher current can flow) is placed on the very top part of the core, as well as the respective pin in the core. Additionally, the access to the sensitive `clk` pin is also facilitated due to the proper spacing between the blocks (since `clk` is not a pin defined for the core itself - only in some of its blocks - it should not be placed towards the edges of the charger core floorplan).

After this, the commands in the TCL script are used to place the pins of the charger core. The pin `pgnd` was placed on the left side, close to `BATCHpower`, since it only needs to be connected to this block. The pins `dgnd` and `dvdd` were placed on the right, to facilitate their connections to different blocks - a similar thinking was applied to the analog pins `vsensbat` and `vbattemp`, placed on the bottom. Additionally, the digital pins `sel[0]`, `sel[1]`, `sel[2]`, `sel[3]` and `en` were placed on the bottom, with a **spacing** of $4.0\mu\text{m}$ - a multiple of $0.4\mu\text{m}$ (as suggested in the Innovus tutorial) - from each other. The positions of other adjacent pins were also selected in order to respect this rule. The **routing** of the charger core is automatically performed with the commands in the TCL script following the positioning of the pins.

```
set init_gnd_net dgnd
set init_lef_file {../lef_libs/header8m2t_V55.lef BATCHARGERbg.lef BATCHARGERctr.lef BATCHARGERpower.lef BATCHARGERsaradc.lef }
set init_oa_search_lib {}
set init_pwr_net dvdd
set init_top_cell BATCHARGERcore
set init_verilog BATCHARGERcore.v

init_design
getIoFlowFlag
setIoFlowFlag 0
floorPlan -site core_2800 -d 640 470.4 5.6 5.6 5.6 5.6
uiSetTool select
getIoFlowFlag

setDrawView fplanso
placeInstance BATCHbg 330 20 R0
placeInstance BATCHctr 350 90 R0
placeInstance BATCHsaradc 20.0 20 R0
placeInstance BATCHpower 20.0 160 R0

setDrawView place
setDrawView fplan
set sprCreateIeRingOffset 1.0
set sprCreateIeRingThreshold 1.0
set sprCreateIeRingJogDistance 1.0
set sprCreateIeRingLayers {}
set sprCreateIeRingOffset 1.0
set sprCreateIeRingThreshold 1.0
set sprCreateIeRingJogDistance 1.0
set sprCreateIeRingLayers {}
set sprCreateIeStripeWidth 10.0
set sprCreateIeStripeThreshold 1.0
set sprCreateIeStripeWidth 10.0
set sprCreateIeStripeThreshold 1.0
set sprCreateIeRingOffset 1.0
set sprCreateIeRingThreshold 1.0
set sprCreateIeRingJogDistance 1.0
set sprCreateIeRingLayers {}
set sprCreateIeStripeWidth 10.0
set sprCreateIeStripeThreshold 1.0

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use ANALOG -pinWidth 78.0 -pinDepth 0.512 -fixOverlap 1 -side Top -layer 4 -assign 335 475.6 -pin iforcedbat
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use ANALOG -pinWidth 2.0 -pinDepth 0.512 -fixOverlap 1 -side Bottom -layer 4 -assign 230 0 -pin vsensbat
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use ANALOG -pinWidth 50.0 -pinDepth 0.512 -fixOverlap 1 -side Left -layer 3 -assign 0 310.2 -pin vin
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -pinWidth 0.4 -pinDepth 0.512 -fixOverlap 1 -side Bottom -layer 4 -spreadType center -spacing 4.0 -pin {{sel[0]} {sel[1]} {sel[2]} {sel[3]} en}
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use ANALOG -pinWidth 2.0 -pinDepth 0.512 -fixOverlap 1 -side Bottom -layer 4 -assign 234 0 -pin vbattemp
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use POWER -pinWidth 2.0 -pinDepth 0.512 -fixOverlap 1 -side Right -layer 3 -assign 645.6 25.0 -pin dvdd
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use GROUND -pinWidth 2.0 -pinDepth 0.512 -fixOverlap 1 -side Right -layer 3 -assign 645.6 30.2 -pin dgnd
setPinAssignMode -pinEditInBatch false

getPinAssignMode -pinEditInBatch -quiet
setPinAssignMode -pinEditInBatch true
editPin -use GROUND -pinWidth 2.0 -pinDepth 0.512 -fixOverlap 1 -side Left -layer 3 -assign 0 160 -pin pgnd
setPinAssignMode -pinEditInBatch false

add_ndr -width {metal1 2.0 metal2 2.0 metal3 2.0 metal4 2.0 metal5 2.0 metal6 2.0 metal7 2.0 metal8 2.0 } -spacing {metal1 0.4 metal2 0.4 metal3 0.4 metal4 0.4 metal5 0.4 metal6 0.4
metal7 0.4 metal8 0.4 } -min_cut {via2 1 via3 1 via4 1 via5 1 via6 1 via7 1 } -add_via {VIAM1M2A VIAM2M3 VIAM3M4 VIAM4M5 VIAM5M6 VIAM6M7 VIAM7M8 genm1m2_w genm1m2a genm1m2b genm2m3_w
genm2m3a genm2m3b genm3m4_w genm3m4a genm3m4b genm4m5_w genm4m5a genm4m5b genm5m6_w genm5m6a genm5m6b genm6m7_w genm6m7a genm6m7b genm7m8_w genm7m8a genm7m8b} -name pwr
setAttribute -net vin -non_default_rule pwr
setAttribute -net iforcedbat -non_default_rule pwr
setAttribute -net pgnd -non_default_rule pwr
setAttribute -net dgnd -non_default_rule pwr
setAttribute -net dvdd -non_default_rule pwr

selectNet dgnd
routeDesign -selected
setSrouteMode -viaConnectToShape { noshape }
sroute -connect { blockPin padPin padRing corePin floatingStripe } -layerChangeRange { metal1(1) metal8(8) } -blockPinTarget { nearestTarget } -padPinPortConnect { allPort oneGeom } -
padPinTarget { nearestTarget } -corePinTarget { firstAfterRowEnd } -floatingStripeTarget { blockring padring ring stripe ringpin blockpin followpin } -allowJogging 1 -
crossoverViaLayerRange { metal1(1) metal8(8) } -nets { dgnd } -allowLayerChange 1 -blockPin useLef -targetViaLayerRange { metal1(1) metal8(8) }
placeAIO
setAttribute -net vrefa -shield_net dgnd
setAttribute -net vrefb -shield_net dgnd
selectNet -shield

setDrawView place
setPlaceMode -fp false
place_design
setNanoRouteMode -quiet -timingEngine {}
setNanoRouteMode -quiet -routeTopRoutingLayer default
setNanoRouteMode -quiet -routeBottomRoutingLayer default
setNanoRouteMode -quiet -drouteEndIteration default
setNanoRouteMode -quiet -routeWithTimingDriven false
setNanoRouteMode -quiet -routeWithSiDriven false
routeDesign -globalDetail

selectWire 249.3000 17.3000 249.5000 26.3000 2 vrefb
uiSetTool copy
editCopy 0.2 0
deselectAll
selectWire 249.7000 17.3000 249.9000 26.3000 2 vrefb
uiSetTool copy
editCopy -0.892 0
deselectAll
selectWire 257.7000 25.3000 257.9000 41.9000 2 vrefa
uiSetTool copy
editCopy -0.362 0
deselectAll
selectWire 257.7000 25.3000 257.9000 41.9000 2 vrefa
uiSetTool copy
editCopy 0.461 0.033
deselectAll
uiSetTool addVia
setEditMode -via_cut_layer via
setEditMode -cut_class {}
setEditMode -via_cell_name 0x0
setEditMode -via_cut_layer via2
editAddVia 257.401 41.403
editAddVia 258.201 40.596
editAddVia 248.997 23.008
editAddVia 249.808 21.794
selectWire 229.7000 19.3000 233.9000 19.5000 3 {tbat[7]}
uiSetTool copy
editCopy 27.522 21.904
deselectAll
uiSetTool copy
selectWire 257.3000 41.3000 261.4200 41.5000 3 vrefa
editCopy 0.814 -0.649
deselectAll
uiSetTool addVia
setEditMode -via_cut_layer via3
setEditMode -cut_class {}
editAddVia 261.01 41.401
editAddVia 261.803 40.596
selectWire 312.1000 0.2600 312.3000 60.3000 4 en
uiSetTool copy
editCopy -51.148 40.98
deselectAll
selectWire 260.9000 41.2400 261.1000 101.2800 4 vrefa
uiSetTool copy
```

```
editCopy 0.035 41.739
deselectAll
uiSetTool addVia
editAddVia 260.997 142.83
selectWire 260.9000 41.2400 261.1000 143.0200 4 vrefa
uiSetTool copy
editCopy 0.871 -0.751
deselectAll
uiSetTool addVia
editAddVia 261.804 142.116
selectWire 257.3000 41.3000 261.1100 41.5000 3 vrefa
uiSetTool copy
editCopy -7.554 -19.557
deselectAll
selectWire 249.7500 21.7000 253.5600 21.9000 3 vrefb
uiSetTool copy
editCopy -0.879 1.222
deselectAll
selectWire 261.7000 40.6000 261.9000 142.2700 4 vrefa
uiSetTool copy
editCopy -8.489 -18.822
deselectAll
selectWire 253.3000 21.7800 253.5000 123.4500 4 vrefa
editCopy 0 19.504
deselectAll
uiSetTool addVia
editAddVia 252.199 22.997
editAddVia 253.405 21.802
selectWire 253.3000 21.7800 253.5000 142.9500 4 vrefa
uiSetTool copy
editCopy -1.028 1.167
deselectAll
uiSetTool addVia
editAddVia 252.201 143.816
editAddVia 253.395 142.788
selectWire 359.2000 460.6000 361.2000 469.7400 4 iforcedbat
uiSetTool copy
editCopy 2 0 -keep_net_name
editCopy 4 0 -keep_net_name
editCopy 6 0 -keep_net_name
editCopy 8 0 -keep_net_name
editCopy 10 0 -keep_net_name
editCopy 12 0 -keep_net_name
editCopy 13 0 -keep_net_name
editCopy -2 0 -keep_net_name
editCopy -4 0 -keep_net_name
editCopy -6 0 -keep_net_name
editCopy -8 0 -keep_net_name
editCopy -10 0 -keep_net_name
editCopy -12 0 -keep_net_name
editCopy -14 0 -keep_net_name
editCopy -16 0 -keep_net_name
editCopy -18 0 -keep_net_name
editCopy -20 0 -keep_net_name
editCopy -22 0 -keep_net_name
editCopy -24 0 -keep_net_name
editCopy -26 0 -keep_net_name
editCopy -28 0 -keep_net_name
editCopy -30 0 -keep_net_name
editCopy -32 0 -keep_net_name
editCopy -34 0 -keep_net_name
editCopy -36 0 -keep_net_name
editCopy -38 0 -keep_net_name
editCopy -40 0 -keep_net_name
editCopy -42 0 -keep_net_name
editCopy -44 0 -keep_net_name
editCopy -46 0 -keep_net_name
editCopy -48 0 -keep_net_name
editCopy -50 0 -keep_net_name
editCopy -52 0 -keep_net_name
editCopy -54 0 -keep_net_name
editCopy -56 0 -keep_net_name
editCopy -58 0 -keep_net_name
editCopy -60 0 -keep_net_name
editCopy -62 0 -keep_net_name
editCopy -63 0 -keep_net_name
deselectAll
selectWire 0.2600 292.8000 18.8000 294.8000 3 vin
uiSetTool copy
editCopy 0 -2 -keep_net_name
editCopy 0 -4 -keep_net_name
editCopy 0 -6 -keep_net_name
editCopy 0 -7.5 -keep_net_name
editCopy 0 2 -keep_net_name
editCopy 0 4 -keep_net_name
editCopy 0 6 -keep_net_name
editCopy 0 8 -keep_net_name
editCopy 0 10 -keep_net_name
editCopy 0 12 -keep_net_name
editCopy 0 14 -keep_net_name
editCopy 0 16 -keep_net_name
editCopy 0 18 -keep_net_name
editCopy 0 20 -keep_net_name
editCopy 0 22 -keep_net_name
editCopy 0 24 -keep_net_name
editCopy 0 26 -keep_net_name
editCopy 0 28 -keep_net_name
editCopy 0 30 -keep_net_name
editCopy 0 32 -keep_net_name
editCopy 0 34 -keep_net_name
editCopy 0 36 -keep_net_name
editCopy 0 38 -keep_net_name
editCopy 0 40 -keep_net_name
editCopy 0 40.5 -keep_net_name
deselectAll
selectWire 16.8000 292.8000 20.2000 294.8000 3 vin
uiSetTool copy
editCopy -1.012 -2 -keep_net_name
editCopy -1.012 -4 -keep_net_name
editCopy -1.012 -6 -keep_net_name
editCopy -1.012 -7.5 -keep_net_name
editCopy -1.012 2 -keep_net_name
editCopy -1.012 4 -keep_net_name
editCopy -1.012 6 -keep_net_name
editCopy -1.012 8 -keep_net_name
editCopy -1.012 10 -keep_net_name
editCopy -1.012 12 -keep_net_name
editCopy -1.012 14 -keep_net_name
editCopy -1.012 16 -keep_net_name
editCopy -1.012 18 -keep_net_name
editCopy -1.012 20 -keep_net_name
editCopy -1.012 22 -keep_net_name
editCopy -1.012 24 -keep_net_name
editCopy -1.012 26 -keep_net_name
editCopy -1.012 28 -keep_net_name
editCopy -1.012 30 -keep_net_name
editCopy -1.012 32 -keep_net_name
editCopy -1.012 34 -keep_net_name
editCopy -1.012 36 -keep_net_name
editCopy -1.012 38 -keep_net_name
editCopy -1.012 40 -keep_net_name
editCopy -1.012 40.5 -keep_net_name
deselectAll

write_lef_abstract -cutObsMinSpacing BATCHARGERcore.lef
```

Figure 1: TCL script `innovus_BATCHARGERcore` that automatically generates the charger core routing and LEF file (with the command `source innovus_BATCHARGERcore`).

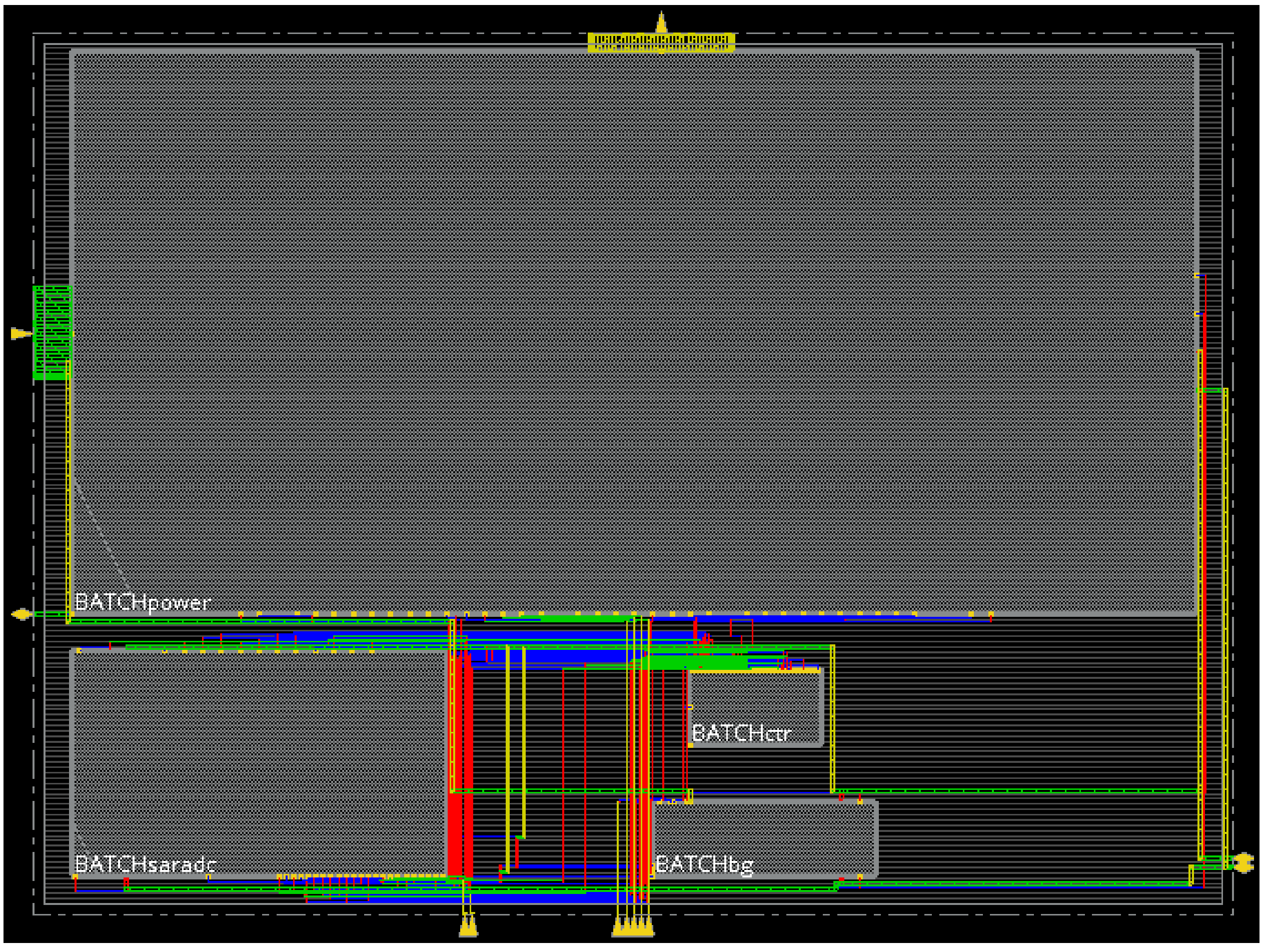

Figure 2: Screenshot of the charger core implemented with Innovus.

Further ahead in the script `innovus_BATCHARGERcore`, the **shielding** of nets `vrefa` and `vrefb` with `dgdn` was supposed to be performed with the commands `setAttribute -net vrefa -shield_net dgnd` and `setAttribute -net vrefb -shield_net dgnd`, respectively. In the 4-bit counter example, the shielding was performed with the analog ground `agnd`; however, this pin is not present in the charger core, thus the digital ground was selected instead. The nets `vrefa` and `vrefb` - from `BATCHbg` - are shielded due to their particular sensitivity. On one hand, `vrefa` is the voltage reference for the power block, while `vrefb` is the voltage reference for the ADC. However, after many devoted efforts in both the script and the GUI interface, it was not possible to perform this task in this way. Thus, this problem was fixed by manually performing the shielding - individually adding proper **wires** and **vias** (in the correct metal layers). The wires were placed with the many commands `selectWire`, `uiSetTool copy` and `editCopy` in the final part of the TCL script shown in Figure 1. On the other hand, the vias were added with the many commands `uiSetTool addVia` and `editAddVia`, also included in this script. The fruitful results of this effort are shown in Figure 3; the connections made to `dgnd` (with vias) can only be seen in Figure 2. It is worth mentioning that rectangles could not simply be added to the design - instead, wires had to copied multiple times in the GUI interface (the respective commands were then copied from the log files to the main TCL script) - since, when the metal layer of these rectangles was properly altered in the GUI, this change was not recorded in the `.cmd` files, thus it would be impossible to automatically generate the shielding in that way.

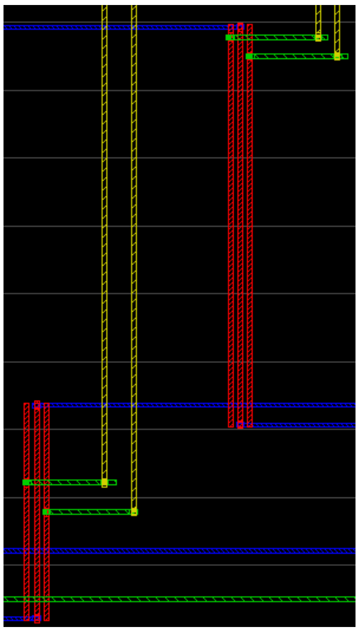

Figure 3: Screenshot of the shielding of `vrefa` and `vrefb` in the charger core.

```
##
## LEF for PtnCells ;
## created by Innovus v20.11-s130_1 on Mon Jan 16 11:25:24 2023
##

VERSION 5.8 ;

BUSBITCHARS "[]" ;
DIVIDERCHAR "/" ;

MACRO BATCHARGERcore
  CLASS BLOCK ;
  SIZE 640.000000 BY 470.000000 ;
  FOREIGN BATCHARGERcore 0.000000 0.000000 ;
  ORIGIN 0 0 ;
  SYMMETRY X Y R90 ;
  PIN iforcedbat
    DIRECTION OUTPUT ;
    USE ANALOG ;
    PORT
      LAYER metal4 ;
        RECT 296.000000 469.480000 374.000000 470.000000 ;
    END
  END iforcedbat
  PIN vsensbat
    DIRECTION INPUT ;
    USE ANALOG ;
    PORT
      LAYER metal4 ;
        RECT 228.800000 0.000000 230.800000 0.520000 ;
    END
  END vsensbat
  PIN vin
    DIRECTION INPUT ;
    USE ANALOG ;
    PORT
      LAYER metal3 ;
        RECT 0.000000 285.200000 0.520000 335.200000 ;
    END
  END vin
  PIN vbattemp
    DIRECTION INPUT ;
    USE ANALOG ;
    PORT
      LAYER metal4 ;
        RECT 232.800000 0.000000 234.800000 0.520000 ;
    END
  END vbattemp
  PIN en
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal4 ;
        RECT 312.000000 0.000000 312.400000 0.520000 ;
    END
  END en
  PIN sel[3]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal4 ;
        RECT 316.000000 0.000000 316.400000 0.520000 ;
    END
  END sel[3]
  PIN sel[2]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal4 ;
        RECT 320.000000 0.000000 320.400000 0.520000 ;
    END
  END sel[2]
  PIN sel[1]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal4 ;
        RECT 324.000000 0.000000 324.400000 0.520000 ;
    END
  END sel[1]
  PIN sel[0]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER metal4 ;
        RECT 328.000000 0.000000 328.400000 0.520000 ;
    END
  END sel[0]
  PIN dvdd
    DIRECTION INOUT ;
    USE POWER ;
    PORT
      LAYER metal3 ;
        RECT 639.480000 24.000000 640.000000 26.000000 ;
    END
  END dvdd
  PIN dgnd
    DIRECTION INOUT ;
    USE GROUND ;
    PORT
      LAYER metal3 ;
        RECT 639.480000 29.200000 640.000000 31.200000 ;
    END
  END dgnd
  PIN pgnd
    DIRECTION INOUT ;
    USE GROUND ;
    PORT
      LAYER metal3 ;
        RECT 0.000000 159.200000 0.520000 161.200000 ;
    END
  END pgnd
  OBS
    LAYER metal1 ;
      RECT 0.000000 0.000000 640.000000 470.000000 ;
    LAYER metal2 ;
      RECT 0.000000 0.000000 640.000000 470.000000 ;
    LAYER metal3 ;
      RECT 0.000000 335.480000 640.000000 470.000000 ;
      RECT 0.800000 284.920000 640.000000 335.480000 ;
      RECT 0.000000 161.480000 640.000000 284.920000 ;
      RECT 0.800000 158.920000 640.000000 161.480000 ;
      RECT 0.000000 31.480000 640.000000 158.920000 ;
      RECT 0.000000 28.920000 639.200000 31.480000 ;
      RECT 0.000000 26.280000 640.000000 28.920000 ;
      RECT 0.000000 23.720000 639.200000 26.280000 ;
      RECT 0.000000 0.000000 640.000000 23.720000 ;
    LAYER metal4 ;
      RECT 374.280000 469.200000 640.000000 470.000000 ;
      RECT 0.000000 469.200000 295.720000 470.000000 ;
      RECT 0.000000 0.800000 640.000000 469.200000 ;
      RECT 235.080000 0.720000 640.000000 0.800000 ;
      RECT 328.600000 0.000000 640.000000 0.720000 ;
      RECT 324.600000 0.000000 327.800000 0.720000 ;
      RECT 320.600000 0.000000 323.800000 0.720000 ;
      RECT 316.600000 0.000000 319.800000 0.720000 ;
      RECT 312.600000 0.000000 315.800000 0.720000 ;
      RECT 235.080000 0.000000 311.800000 0.720000 ;
      RECT 231.080000 0.000000 232.520000 0.800000 ;
      RECT 0.000000 0.000000 228.520000 0.800000 ;
    LAYER metal5 ;
      RECT 0.000000 0.000000 640.000000 470.000000 ;
    LAYER metal6 ;
      RECT 0.000000 0.000000 640.000000 470.000000 ;
    LAYER metal7 ;
      RECT 0.000000 0.000000 640.000000 470.000000 ;
    LAYER metal8 ;
      RECT 0.000000 0.000000 640.000000 470.000000 ;
  END
END BATCHARGERcore

END LIBRARY
```

Figure 4: LEF file `BATCHARGERcore.lef` of the charger core.

Another very relevant topic to mention is the interconnections width defined for some of the pins in the charger core. The most relevant one is perhaps `iforcedbat`, relative to the output current to the battery. As seen in Lab 2, the maximum value of the battery current was about $\mathbf{200mA}$ - which occurs in constant current (CC) mode. According to the DC rules of electromigration specified in the IO library documents (included in the class slides of Chapter 8), the maximum value of $\mathrm{I_{max\_dc}[mA/\mu m]}$ is 2.56 for most metals (M1 to M6) and 8 for higher metals (M7 and M8). In the charger core, only the first case applies for `iforcedbat`. Thus, this leads to a width of $\approx 78\mu\mathrm{m}$. For this purpose, wires of layer `metal4` (layer used in the respective pin of `BATCHpower`) were copied close to each other, similarly to what was performed in the shielding (thus, the “copy” commands in the TCL script with wires of `iforcedbat`). By doing this, the desired width for the interconnection was obtained - this is shown on the very top of Figure 2, in yellow. Another important pin to take into account is `vin` (input voltage) - whose pin, in the available LEF file for the power block, is defined with a width of $50\mu\mathrm{m}$. For this purpose, an analogous procedure was made in order to obtain an interconnection of $50\mu\mathrm{m}$ in `metal3`, as shown on the left side of Figure 2 (the pin `vin` was placed there to facilitate this wide connection with the `BATCHpower` block). The configuration of these interconnections was once again performed this way - not with single commands in the TCL script - due to limitations in the usage of the Innovus tool. The remaining interconnections in the core were defined according to the previously defined widths in the original TCL script and the widths of all other pins in the design (much lower, when compared to `iforcedbat` and `vin`). In the end, the **LEF** file is generated with the final command `write_lef_abstract -cutObsMinSpacing BATCHARGERcore.lef` in the TCL script, from which the file shown in Figure 4 is obtained.

3 Charger top

For the IO ring design, the file `BATCHARGER.save.io` was defined with the respective placement of the cells in the full chip. A **corner cell** is placed in each corner of the design, with the proper orientation for the correct correspondence with the other cells in the IO ring. The corresponding cell name is “CORNERHB”, as defined in `foc0h_a33_t33_generic_io.8m2t.lef`. These cells are for a **core limited design**, the most appropriate in this case, due to the considerable size of the charger core (as seen in Figure 8). On the left, the instances `inst1` to `inst4` (corresponding to **digital IOs**) are placed. As seen in the `BATCHARGER.v` file (shown in Figure 6), the cells XMHB (programmable 3.3V input buffer, core limit) were used here. On the right side, the cells ULSCI0CUTHB - now defined in the LEF file `foc0h_a33_t12_analogesd_io.8m2t.lef` - are used for **analog IOs**; moreover the **analog ground** and **analog supply** were also included, with the cells GNDACUTHB (ground pad for analog block) and VCC12ACUTHB (3.3V power pad for analog block), respectively. On the top, the **core ground** and **core supply** correspond to the respective IO cells GNDKHB and VCCKHB (in a core limited design, ground and power supply for internal core cells and I/O to core interface, respectively), whereas the instance `instrc` corresponds to a **right cut** (cell RCUT12HB). Finally, a **left cut** (instance `instlc`) was placed in the bottom, along with the cells for the **IOs ground** and **supply** (respectively, GND3IOHB and VCC3IOHB - ground and power supply for 3.3V input/output buffers) and the other digital IO (instance `inst5`). The right and left cuts were placed this way in order to separate IO domains (digital and analog). In these various cells, the inputs, outputs and input-outputs defined for the module `BATCHARGER` were attributed as shown in Figure 6.

In order to perform automatic routing of the charger top, the TCL script `innovus_BATCHARGER` - included in Figure 7 - was developed. Initially, commands `set init_lef_file` are used to call the IO libraries LEF files, as well as the charger core LEF file obtained previously. Moreover, similar commands are used for the `BATCHARGER.save.io` and `BATCHARGER.v` files. After this, the floorplan is created with a width $1250\mu\mathrm{m}$ and a height $1001.6\mu\mathrm{m}$ (multiples of $0.4\mu\mathrm{m}$ and $3.2\mu\mathrm{m}$, respectively), taking into account the size of the charger core, which is placed approximately in the middle of the design; core to IO margins of $6.0\mu\mathrm{m}$ (multiples of $0.4\mu\mathrm{m}$) were also selected. After defining the metal widths and spacings, global nets are connected and **fill cells** EMPTY8HB, EMPTY4HB, EMPTY2HB and EMPTY1HB (for different number of grids) are added. These four types of fill cells (defined in `foc0h_a33_t33_generic_io.8m2t.lef`) are used in core limited designs. Once this is done, the routing is performed similarly to the charger core. By running the command `source innovus_BATCHARGER` in a newly created directory, the results shown in Figures 8 to 10 are obtained.

```
###############################################################
#  Generated by:      Cadence Innovus 20.11-s130_1
#  OS:                Linux x86_64(Host ID fatima.novalocal)
#  Design:            BATCHARGER
#  Command:           saveIoFile -locations BATCHARGER.save.io
###############################################################

(globals
    version = 3
    io_order = default
)
(iopad
    (topright
        (inst name="CornerCell2"        orientation=R90 cell="CORNERHB" )
    )
    (top
        (inst  name="gndinst1" )
        (inst  name="vccinst1" )
        (inst  name="instrc" )
    )
    (topleft
        (inst name="CornerCell1"        orientation=R180 cell="CORNERHB" )
    )
    (left
        (inst  name="inst1" )
        (inst  name="inst2" )
        (inst  name="inst3" )
        (inst  name="inst4" )
    )
    (bottomleft
        (inst name="CornerCell4"        orientation=R270 cell="CORNERHB" )
    )
    (bottom
        (inst  name="gndio" )
        (inst  name="vcc3io" )
        (inst  name="inst5" )
        (inst  name="instlc" )
    )
    (bottomright
        (inst name="CornerCell3"        orientation=R0 cell="CORNERHB" )
    )
    (right
        (inst  name="vccinst2")
        (inst  name="analogpad1" )
        (inst  name="analogpad2" )
        (inst  name="analogpad3" )
        (inst  name="gndinst2" )
    )
)
```

Figure 5: File `BATCHARGER.save.io` which includes the relative positions of the IO cells.

```
module BATCHARGER ( iforcedbat, vsensbat, vin, vbattemp, en, sel, dvdd, dgnd, pgnd);
   output iforcedbat;
   input vsensbat, vin, vbattemp, en;
   inout dvdd, dgnd, pgnd;
   input [3:0] sel;

   wire pu, pd, smt, en, e2, e4, sr;
   wire [3:0]  sel_core;

   assign pu = 1'b0;
   assign pd = 1'b0;
   assign smt = 1'b0;
   assign en = 1'b1;
   assign e2 = 1'b0;
   assign e4 = 1'b0;
   assign sr = 1'b1;

   BATCHARGERcore   BATCH (.en(en_core), .iforcedbat(iforcedbat), .vsensbat(vsensbat) , .vin(vin), .vbattemp(vbattemp), .sel(sel_core), .dvdd(dvdd), .dgnd(dgnd), .pgnd(pgnd));

   GNDKHB gndinst1 (.GND(dgnd));
   VCCKHB vccinst1 (.VCC(dvdd));
   RCUT12HB instrc();

   GNDACUTHB gndinst2 (.GNDANA(pgnd));
   ULSCI0CUTHB analogpad3 (.O(vbattemp));
   ULSCI0CUTHB analogpad2 (.O(vsensbat));
   ULSCI0CUTHB analogpad1 (.O(iforcedbat));
   VCC12ACUTHB vccinst2 (.VCC12ANA(vin));

   LCUT12HB instlc();
   XMHB inst5 (.O(sel_core[3]), .I(sel[3]), .PU(pu), .PD(pd), .SMT(smt));
   VCC3IOHB vcc3io ;
   GND3IOHB gndio ;

   XMHB inst1 (.O(en_core), .I(en), .PU(pu), .PD(pd), .SMT(smt));
   XMHB inst2 (.O(sel_core[0]), .I(sel[0]), .PU(pu), .PD(pd), .SMT(smt));
   XMHB inst3 (.O(sel_core[1]), .I(sel[1]), .PU(pu), .PD(pd), .SMT(smt));
   XMHB inst4 (.O(sel_core[2]), .I(sel[2]), .PU(pu), .PD(pd), .SMT(smt));

endmodule
```

Figure 6: File `BATCHARGER.v` which includes the IO cells used for the charger top.

```
set init_gnd_net dgnd
set init_io_file BATCHARGER.save.io
set init_lef_file {../lef_libs/header8m2t_V55.lef ../BATCHARGERcore_v_lefs/BATCHARGERcore.lef foc0h_a33_t33_generic_io.8m2t.lef}
set init_oa_search_lib {}
set init_original_verilog_files BATCHARGER.v
set init_pwr_net dvdd
set init_top_cell BATCHARGER
set init_verilog BATCHARGER.v
set init_lef_file {../lef_libs/header8m2t_V55.lef ../BATCHARGERcore_v_lefs/BATCHARGERcore.lef foc0h_a33_t33_generic_io.8m2t.lef  foc0h_a33_t12_analogesd_io.8m2t.lef}

init_design
floorPlan -site core_2800 -d 1250 1001.6 6.0 6.0 6.0 6.0
placeInstance BATCH 310 265

add_ndr -width {metal1 2.0 metal2 2.0 metal3 2.0 metal4 2.0 metal5 2.0 metal6 2.0 metal7 2.0 metal8 2.0 } -spacing {metal1 0.4 metal2 0.4 metal3 0.4 metal4 0.4 metal5 0.4 metal6 0.4 metal7 0.4 metal8 0.4 } -min_cut {via2 1 via3 1 via4 1 via5 1 via6 1
via7 1 } -add_via {VIAM1M2A VIAM2M3 VIAM3M4 VIAM4M5 VIAM5M6 VIAM6M7 VIAM7M8 genm1m2_w genm1m2a genm1m2b genm2m3_w genm2m3a genm2m3b genm3m4_w genm3m4a genm3m4b genm4m5_w genm4m5a genm4m5b genm5m6_w genm5m6a genm5m6b genm6m7_w genm6m7a genm6m7b
genm7m8_w genm7m8a genm7m8b} -name pwr
setAttribute -net dgnd -non_default_rule pwr
setAttribute -net dvdd -non_default_rule pwr
setAttribute -net pgnd -non_default_rule pwr
setAttribute -net iforcedbat -non_default_rule pwr
setAttribute -net vin -non_default_rule pwr

clearGlobalNets
globalNetConnect dvdd -type tiehi -instanceBasename *
globalNetConnect dgnd -type tielo -instanceBasename *
addIoFiller -cell EMPTY8HB
addIoFiller -cell EMPTY4HB
addIoFiller -cell EMPTY2HB
addIoFiller -cell EMPTY1HB

globalNetConnect dgnd -type pgpin -pin GND -instanceBasename * -hierarchicalInstance {}
globalNetConnect dvdd -type pgpin -pin VCC -instanceBasename * -hierarchicalInstance {}

setAttribute -net vsensbat -shield_net dgnd
selectNet -shield
routeDesign -selected
routeDesign -globalDetail

setSrouteMode -viaConnectToShape { noshape }
sroute -connect { blockPin padPin padRing corePin floatingStripe } -layerChangeRange { metal1(1) metal8(8) } -blockPinTarget { nearestTarget } -padPinPortConnect { allPort oneGeom } -padPinTarget { nearestTarget } -corePinTarget { firstAfterRowEnd } -
floatingStripeTarget { blockring padring ring stripe ringpin blockpin followpin } -allowJogging 1 -crossoverViaLayerRange { metal1(1) metal8(8) } -nets { dvdd } -allowLayerChange 1 -blockPin useLef -targetViaLayerRange { metal1(1) metal8(8) }
setSrouteMode -viaConnectToShape { noshape }
sroute -connect { blockPin padPin padRing corePin floatingStripe } -layerChangeRange { metal1(1) metal8(8) } -blockPinTarget { nearestTarget } -padPinPortConnect { allPort oneGeom } -padPinTarget { nearestTarget } -corePinTarget { firstAfterRowEnd } -
floatingStripeTarget { blockring padring ring stripe ringpin blockpin followpin } -allowJogging 1 -crossoverViaLayerRange { metal1(1) metal8(8) } -nets { dgnd } -allowLayerChange 1 -blockPin useLef -targetViaLayerRange { metal1(1) metal8(8) }
routeDesign -selected
setPlaceMode -fp false
place_design

setNanoRouteMode -quiet -timingEngine {}
setNanoRouteMode -quiet -routeTopRoutingLayer default
setNanoRouteMode -quiet -routeBottomRoutingLayer default
setNanoRouteMode -quiet -drouteEndIteration default
setNanoRouteMode -quiet -routeWithTimingDriven false
setNanoRouteMode -quiet -routeWithSiDriven false
routeDesign -globalDetail

selectWire 1091.2000 400.6000 1093.2000 738.0000 6 iforcedbat
uiSetTool copy
editCopy -2 0 -keep_net_name
editCopy -4 0 -keep_net_name
editCopy -6 0 -keep_net_name
editCopy -8 0 -keep_net_name
editCopy -10 0 -keep_net_name
editCopy -12 0 -keep_net_name
editCopy -14 0 -keep_net_name
editCopy -16 0 -keep_net_name
editCopy -18 0 -keep_net_name
editCopy -20 0 -keep_net_name
editCopy -22 0 -keep_net_name
editCopy -24 0 -keep_net_name
editCopy -26 0 -keep_net_name
editCopy -28 0 -keep_net_name
editCopy -30 0 -keep_net_name
editCopy -32 0 -keep_net_name
editCopy -34 0 -keep_net_name
editCopy -36 0 -keep_net_name
editCopy -38 0 -keep_net_name
editCopy -40 0 -keep_net_name
editCopy -42 0 -keep_net_name
editCopy -44 0 -keep_net_name
editCopy -46 0 -keep_net_name
editCopy -48 0 -keep_net_name
editCopy -50 0 -keep_net_name
editCopy -52 0 -keep_net_name
editCopy -54 0 -keep_net_name
editCopy -56 0 -keep_net_name
editCopy -58 0 -keep_net_name
editCopy -60 0 -keep_net_name
editCopy -62 0 -keep_net_name
editCopy -64 0 -keep_net_name
editCopy -66 0 -keep_net_name
editCopy -68 0 -keep_net_name
editCopy -70 0 -keep_net_name
editCopy -72 0 -keep_net_name
editCopy -74 0 -keep_net_name
editCopy -76 0 -keep_net_name
editCopy -78 0 -keep_net_name
deselectAll
selectWire 647.2000 736.4000 1093.2000 738.4000 5 iforcedbat
editCopy 0 2 -keep_net_name
editCopy 0 4 -keep_net_name
editCopy 0 6 -keep_net_name
editCopy 0 8 -keep_net_name
editCopy 0 10 -keep_net_name
editCopy 0 12 -keep_net_name
editCopy 0 14 -keep_net_name
editCopy 0 16 -keep_net_name
editCopy 0 18 -keep_net_name
editCopy 0 20 -keep_net_name
editCopy 0 22 -keep_net_name
editCopy 0 24 -keep_net_name
editCopy 0 26 -keep_net_name
editCopy 0 28 -keep_net_name
editCopy 0 30 -keep_net_name
editCopy 0 32 -keep_net_name
editCopy 0 34 -keep_net_name
editCopy 0 36 -keep_net_name
editCopy 0 38 -keep_net_name
editCopy 0 40 -keep_net_name
editCopy 0 42 -keep_net_name
editCopy 0 44 -keep_net_name
editCopy 0 46 -keep_net_name
editCopy 0 48 -keep_net_name
editCopy 0 50 -keep_net_name
editCopy 0 52 -keep_net_name
editCopy 0 54 -keep_net_name
editCopy 0 56 -keep_net_name
editCopy 0 58 -keep_net_name
editCopy 0 60 -keep_net_name
editCopy 0 62 -keep_net_name
editCopy 0 64 -keep_net_name
editCopy 0 66 -keep_net_name
editCopy 0 68 -keep_net_name
editCopy 0 70 -keep_net_name
editCopy 0 72 -keep_net_name
editCopy 0 74 -keep_net_name
editCopy 0 76 -keep_net_name
editCopy 0 78 -keep_net_name
deselectAll
selectWire 307.2000 250.8000 1086.8000 252.8000 5 vin
uiSetTool copy
editCopy 0 -2 -keep_net_name
editCopy 0 -4 -keep_net_name
editCopy 0 -6 -keep_net_name
editCopy 0 -8 -keep_net_name
editCopy 0 -10 -keep_net_name
editCopy 0 -12 -keep_net_name
editCopy 0 -14 -keep_net_name
editCopy 0 -16 -keep_net_name
editCopy 0 -18 -keep_net_name
editCopy 0 -20 -keep_net_name
editCopy 0 -22 -keep_net_name
editCopy 0 -24 -keep_net_name
editCopy 0 -26 -keep_net_name
editCopy 0 -28 -keep_net_name
editCopy 0 -30 -keep_net_name
editCopy 0 -32 -keep_net_name
editCopy 0 -34 -keep_net_name
editCopy 0 -36 -keep_net_name
editCopy 0 -38 -keep_net_name
editCopy 0 -40 -keep_net_name
editCopy 0 -42 -keep_net_name
editCopy 0 -44 -keep_net_name
editCopy 0 -46 -keep_net_name
editCopy 0 -48 -keep_net_name
editCopy 0 -50 -keep_net_name
deselectAll
selectWire 306.8000 250.8000 308.8000 556.0000 4 vin
uiSetTool copy
editCopy -2 0 -keep_net_name
editCopy -4 0 -keep_net_name
editCopy -6 0 -keep_net_name
editCopy -8 0 -keep_net_name
editCopy -10 0 -keep_net_name
editCopy -12 0 -keep_net_name
editCopy -14 0 -keep_net_name
editCopy -16 0 -keep_net_name
editCopy -18 0 -keep_net_name
editCopy -20 0 -keep_net_name
editCopy -22 0 -keep_net_name
editCopy -24 0 -keep_net_name
editCopy -26 0 -keep_net_name
editCopy -28 0 -keep_net_name
editCopy -30 0 -keep_net_name
editCopy -32 0 -keep_net_name
editCopy -34 0 -keep_net_name
editCopy -36 0 -keep_net_name
editCopy -38 0 -keep_net_name
editCopy -40 0 -keep_net_name
editCopy -42 0 -keep_net_name
editCopy -44 0 -keep_net_name
editCopy -46 0 -keep_net_name
editCopy -48 0 -keep_net_name
editCopy -50 0 -keep_net_name
deselectAll
selectWire 1091.2000 400.6000 1093.2000 738.0000 6 iforcedbat
uiSetTool copy
editCopy 0 -43.68 -keep_net_name
editCopy -2 -43.68 -keep_net_name
editCopy -4 -43.68 -keep_net_name
editCopy -6 -43.68 -keep_net_name
editCopy -8 -43.68 -keep_net_name
editCopy -10 -43.68 -keep_net_name
editCopy -12 -43.68 -keep_net_name
editCopy -14 -43.68 -keep_net_name
editCopy -16 -43.68 -keep_net_name
editCopy -18 -43.68 -keep_net_name
editCopy -20 -43.68 -keep_net_name
editCopy -22 -43.68 -keep_net_name
editCopy -24 -43.68 -keep_net_name
editCopy -26 -43.68 -keep_net_name
editCopy -28 -43.68 -keep_net_name
editCopy -30 -43.68 -keep_net_name
editCopy -32 -43.68 -keep_net_name
editCopy -34 -43.68 -keep_net_name
editCopy -36 -43.68 -keep_net_name
editCopy -38 -43.68 -keep_net_name
editCopy -40 -43.68 -keep_net_name
editCopy -42 -43.68 -keep_net_name
editCopy -44 -43.68 -keep_net_name
editCopy -46 -43.68 -keep_net_name
editCopy -48 -43.68 -keep_net_name
editCopy -50 -43.68 -keep_net_name
editCopy -52 -43.68 -keep_net_name
editCopy -54 -43.68 -keep_net_name
editCopy -56 -43.68 -keep_net_name
editCopy -58 -43.68 -keep_net_name
editCopy -60 -43.68 -keep_net_name
editCopy -62 -43.68 -keep_net_name
editCopy -64 -43.68 -keep_net_name
editCopy -66 -43.68 -keep_net_name
editCopy -68 -43.68 -keep_net_name
editCopy -70 -43.68 -keep_net_name
editCopy -72 -43.68 -keep_net_name
editCopy -74 -43.68 -keep_net_name
editCopy -76 -43.68 -keep_net_name
editCopy -78 -43.68 -keep_net_name
deselectAll
selectWire 647.2000 736.4000 1093.2000 738.4000 5 iforcedbat
uiSetTool copy
editCopy -68.583 0 -keep_net_name
editCopy -68.583 2 -keep_net_name
editCopy -68.583 4 -keep_net_name
editCopy -68.583 6 -keep_net_name
editCopy -68.583 8 -keep_net_name
editCopy -68.583 10 -keep_net_name
editCopy -68.583 12 -keep_net_name
editCopy -68.583 14 -keep_net_name
editCopy -68.583 16 -keep_net_name
editCopy -68.583 18 -keep_net_name
editCopy -68.583 20 -keep_net_name
editCopy -68.583 22 -keep_net_name
editCopy -68.583 24 -keep_net_name
editCopy -68.583 26 -keep_net_name
editCopy -68.583 28 -keep_net_name
editCopy -68.583 30 -keep_net_name
editCopy -68.583 32 -keep_net_name
editCopy -68.583 34 -keep_net_name
editCopy -68.583 36 -keep_net_name
editCopy -68.583 38 -keep_net_name
editCopy -68.583 40 -keep_net_name
editCopy -68.583 42 -keep_net_name
editCopy -68.583 44 -keep_net_name
editCopy -68.583 46 -keep_net_name
editCopy -68.583 48 -keep_net_name
editCopy -68.583 50 -keep_net_name
editCopy -68.583 52 -keep_net_name
editCopy -68.583 54 -keep_net_name
editCopy -68.583 56 -keep_net_name
editCopy -68.583 58 -keep_net_name
editCopy -68.583 60 -keep_net_name
editCopy -68.583 62 -keep_net_name
editCopy -68.583 64 -keep_net_name
editCopy -68.583 66 -keep_net_name
editCopy -68.583 68 -keep_net_name
editCopy -68.583 70 -keep_net_name
editCopy -68.583 72 -keep_net_name
editCopy -68.583 74 -keep_net_name
editCopy -68.583 76 -keep_net_name
editCopy -68.583 78 -keep_net_name
deselectAll
selectWire 306.8000 250.8000 308.8000 556.0000 4 vin
uiSetTool copy
editCopy 0 78.482 -keep_net_name
editCopy -2 78.482 -keep_net_name
editCopy -4 78.482 -keep_net_name
editCopy -6 78.482 -keep_net_name
editCopy -8 78.482 -keep_net_name
editCopy -10 78.482 -keep_net_name
editCopy -12 78.482 -keep_net_name
editCopy -14 78.482 -keep_net_name
editCopy -16 78.482 -keep_net_name
editCopy -18 78.482 -keep_net_name
editCopy -20 78.482 -keep_net_name
editCopy -22 78.482 -keep_net_name
editCopy -24 78.482 -keep_net_name
editCopy -26 78.482 -keep_net_name
editCopy -28 78.482 -keep_net_name
editCopy -30 78.482 -keep_net_name
editCopy -32 78.482 -keep_net_name
editCopy -34 78.482 -keep_net_name
editCopy -36 78.482 -keep_net_name
editCopy -38 78.482 -keep_net_name
editCopy -40 78.482 -keep_net_name
editCopy -42 78.482 -keep_net_name
editCopy -44 78.482 -keep_net_name
editCopy -46 78.482 -keep_net_name
editCopy -48 78.482 -keep_net_name
editCopy -50 78.482 -keep_net_name
deselectAll
selectWire 307.2000 250.8000 1086.8000 252.8000 5 vin
uiSetTool copy
editCopy -50.355 0 -keep_net_name
editCopy -50.355 -2 -keep_net_name
editCopy -50.355 -4 -keep_net_name
editCopy -50.355 -6 -keep_net_name
editCopy -50.355 -8 -keep_net_name
editCopy -50.355 -10 -keep_net_name
editCopy -50.355 -12 -keep_net_name
editCopy -50.355 -14 -keep_net_name
editCopy -50.355 -16 -keep_net_name
editCopy -50.355 -18 -keep_net_name
editCopy -50.355 -20 -keep_net_name
editCopy -50.355 -22 -keep_net_name
editCopy -50.355 -24 -keep_net_name
editCopy -50.355 -26 -keep_net_name
editCopy -50.355 -28 -keep_net_name
editCopy -50.355 -30 -keep_net_name
editCopy -50.355 -32 -keep_net_name
editCopy -50.355 -34 -keep_net_name
editCopy -50.355 -36 -keep_net_name
editCopy -50.355 -38 -keep_net_name
editCopy -50.355 -40 -keep_net_name
editCopy -50.355 -42 -keep_net_name
editCopy -50.355 -44 -keep_net_name
editCopy -50.355 -46 -keep_net_name
editCopy -50.355 -48 -keep_net_name
editCopy -50.355 -50 -keep_net_name
deselectAll
selectWire 1084.8000 256.8000 1098.6000 258.8000 5 vin
uiSetTool copy
editCopy -0.606 -2 -keep_net_name
editCopy -0.606 -4 -keep_net_name
editCopy -0.606 -6 -keep_net_name
editCopy -0.606 -8 -keep_net_name
editCopy -0.606 -10 -keep_net_name
editCopy -0.606 -12 -keep_net_name
editCopy -0.606 -14 -keep_net_name
editCopy -0.606 -16 -keep_net_name
editCopy -0.606 -18 -keep_net_name
editCopy -0.606 -20 -keep_net_name
editCopy -0.606 -22 -keep_net_name
editCopy -0.606 -24 -keep_net_name
editCopy -0.606 -26 -keep_net_name
editCopy -0.606 -28 -keep_net_name
editCopy -0.606 -30 -keep_net_name
editCopy -0.606 -32 -keep_net_name
deselectAll
```

Figure 7: TCL script `innovus_BATCHARGER` that automatically generates the charger top routing (with the command `source innovus_BATCHARGER`).

In this case, the interconnections widths were defined similarly to the charger core. The widths for most pins are defined accordingly to the respective pin widths. However, once again, an interconnection width of $\approx 78\mu m$ should be used for `iforcedbat`, along with $50\mu m$ for `vin` - as explained in section 2. Once again, it was not possible to do this initially with the TCL script, thus rectangles were copied into the proper positions in order to obtain the result included in Figure 8 - thus the several `selectWire`, `uiSetTool copy` and `editCopy` commands shown in Figure 7. As seen below, the process was successfully performed (this can be also confirmed in the log report `innovus_BATCHARGER.log` included in the submitted zip file) and the IO cells were positioned as indicated in `BATCHARGER.save.io`. An important aspect to mention is the **trade-off** that exists between area and performance in the floorplan generation. In order to use very high widths for certain interconnections, the performance of the circuit improves - since, for instance, the risk of electromigration (which can cause open circuits) is much lower. However, to do this, it was necessary to increase the area of the floorplan, defined in `innovus_BATCHARGER`; moreover, this can lead to more power consumption. Higher metals - with higher current densities (M7 and M8) - could be used instead (to have lower widths, since these metal layers are thicker); in fact, very high widths in lower metals can lead to an unwanted topography in the chip, thus causing problems in nanofabrication.

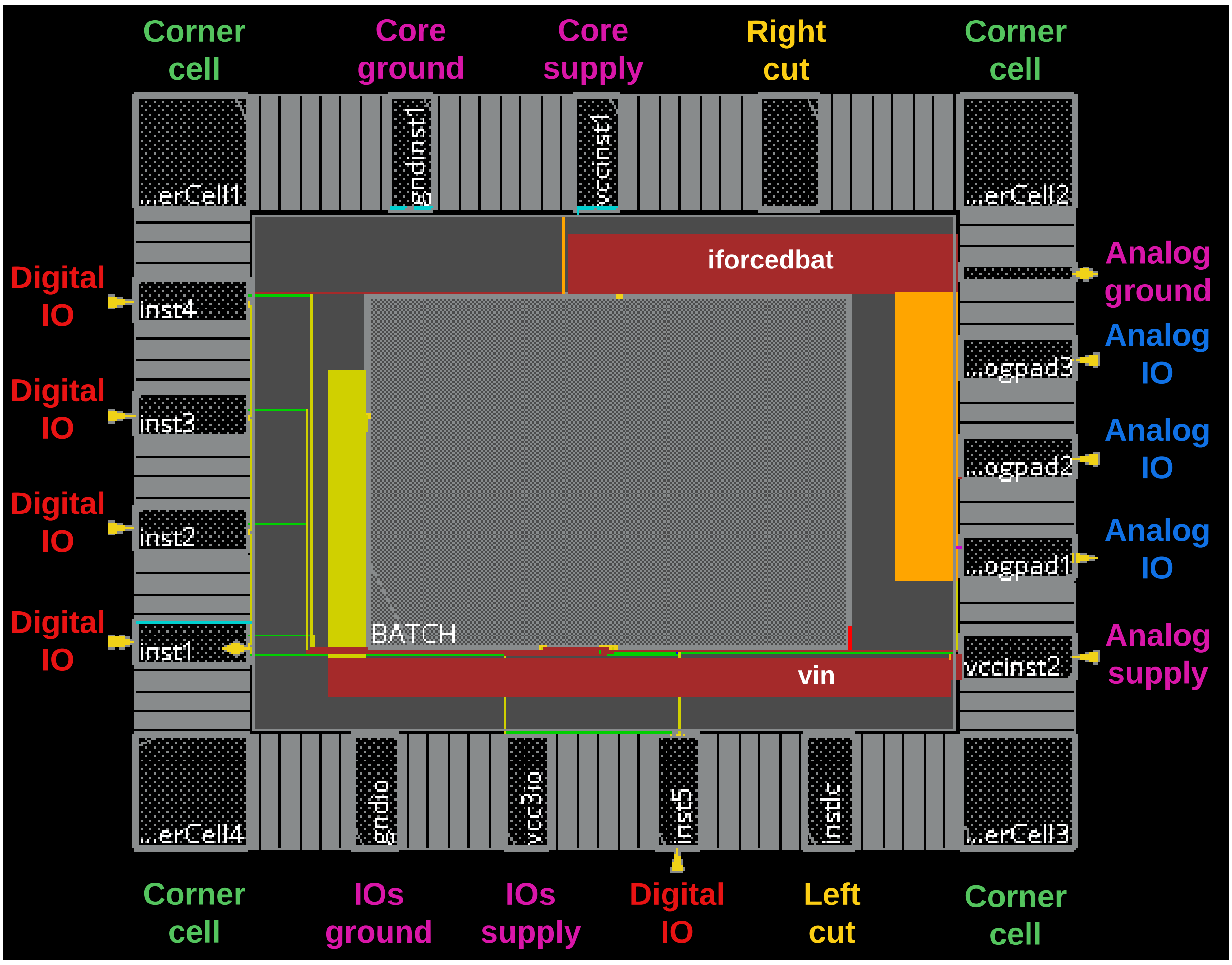

Figure 8: Screenshot of the charger top with the respective IO cells indicated.

Regarding **shielding**, this process was performed in the charger core for the sensitive nets `vrefa` and `vrefb`. However, these are internal pins to the core, thus they cannot be shielded in the full chip design. Taking into account all the pins included in `BATCHARGER.v`, shielding was performed with `vsensbat` (the sensed voltage in the battery), which seems to be the most sensitive net in this case. For this purpose, the proper commands were included in the TCL script (right after defining the global nets) to shield `vsensbat` with `dgnd` (since `dgnd` was also used for shielding in the charger core). However, in this case, the process was successfully performed automatically with the script, thus it wasn't necessary to manually add wires and vias. In fact, the results of this process are shown in Figure 9 and in the generated file included in Figure 10.

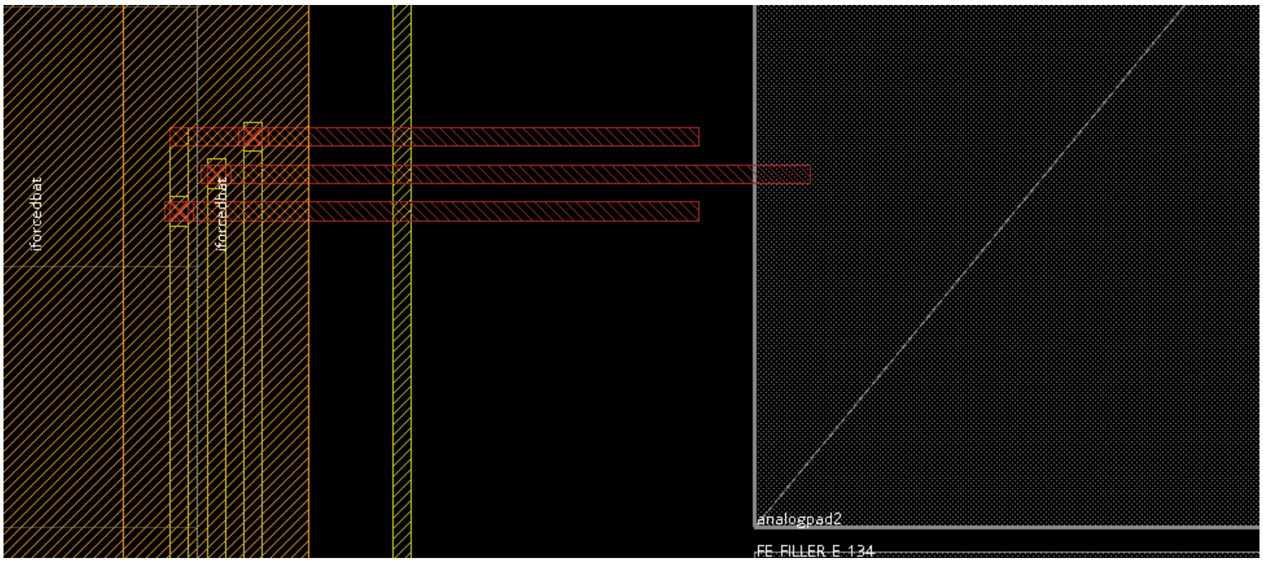

Figure 9: Screenshot of the shielding of `vsensbat` in the charger top.

```
--------------------------------------------------------------------------------
Name     : Shielded net name
Length   : Shielded net length
Shield   : Total length of shielding wire
Ratio    : Average shielding ratio
--------------------------------------------------------------------------------
Name           Length     Shield     Ratio
                                             Layer:       Length      Shield  Ratio
--------------------------------------------------------------------------------
vsensbat:
                792.7     1577.0     0.995
                                             metal3:       551.6      1099.2  0.996
                                             metal4:       234.7       467.4  0.996
                                             metal5:         6.4        10.4  0.812
--------------------------------------------------------------------------------

 Average ratio:                      0.995
                                             metal3:         551.6      1099.2  0.996
                                             metal4:         234.7       467.4  0.996
                                             metal5:           6.4        10.4  0.812
```

Figure 10: Shielding analysis results obtained in the file `BATCHARGER_init_shield.rpt`.

4 Conclusion

In this laboratory assignment, a valuable insight into SoC floorplanning and IO ring design has been acquired, using the Innovus tool. Initially, the floorplan of the battery charger core was developed according to the proper specifications and optimization procedures. The blocks were placed in order to separate a noisy block from a sensitive area and the pin access to high current and sensitive pins was facilitated. In addition, the interconnection widths were accordingly defined - for instance, by taking into account the pin widths and the effect of electromigration - and the most sensitive nets were shielded with ground. Using a TCL script, the charger core was automatically generated and the respective LEF file was created. With another script, the IO ring core limited design was developed for the charger top. The IO cells selected in this design included buffering of digital signals, pass through for analog signals, supply/ground for core power domains, supply/ground for IO cells, cut cells to separate IO domains, corner cells and fill cells. By complying with the technology IO rules, automatic routing and shielding were successfully performed. Thus, all of the proposed objectives for this laboratory assignment have been achieved.