

ΨΗΦΙΑΚΑ ΟΛΟΚΛΗΡΩΜΕΝΑ ΚΥΚΛΩΜΑΤΑ VLSI-ASIC ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ

Νικόλαος Γιαννόπουλος
ΑΕΜ: 9629
Email: Ngiannop@ece.auth.gr

Πίνακας περιεχομένων

Εισαγωγή.....	3
Άσκηση 1.....	3
Άσκηση 2.....	30
Άσκηση 3.....	35
Άσκηση 4.....	42
Άσκηση 5.....	44
Άσκηση 6.....	46
Άσκηση 7.....	48
Άσκηση 8.....	53
Άσκηση 9.....	55

Figure 1 Exercise 1 Report Clock Gating	4
Figure 2 Exercise 1 Top level Module	4
Figure 3 Exercise 1 Check Timing Intent	5
Figure 4 Exercise 1 FloorPlan	7
Figure 5 Exercise 1 Results of FloorPlan	7
Figure 6 Exercise 1 Rings & Stripes and output	8
Figure 7 Exercise 1 Pins Vdd/GND	9
Figure 8 Exercise 1 SRoute & results.....	10
Figure 9 Exercise 1 Place I/O results	11
Figure 10 Exercise 1 Check Placement.....	12
Figure 11 Exercise 1 Early Global Route.....	13
Figure 12 Exercise 1 Timing Analysis with Pre-CTS and Setup Time	13
Figure 13 Exercise 1 Results setup time.....	14
Figure 14 Exercise 1 Timing Analysis with Pre-CTS and Hold Time.....	14
Figure 15 Exercise 1 Results hold time	15
Figure 16 Exercise 1 Set Power Analysis Mode step 1.....	15
Figure 17 Exercise 1 Set Power Analysis Mode step 2.....	16
Figure 18 Exercise 1 Set Power Analysis Mode step 3 & Power Rail Plot.....	16
Figure 19 Exercise 1 Mode Setup Optimization.....	17
Figure 20 Exercise 1 SRoute All levels of metals	18
Figure 21 Exercise 1 Early Global Route all metals & results	18
Figure 22 Exercise 1 SRoute Metal2-Metal10 levels of metals & Results.....	19
Figure 23 Exercise 1 Create Non Default Rules.....	20
Figure 24 Exercise 1 Clock Tree Debugger	21
Figure 25 Exercise 1 Example of Clock Tree Debugger	21
Figure 26 Exercise 1 Optimazation PostCTS.....	22
Figure 27 Exercise 1 Tlming Analysis Setup/Hold Time Post-CTS	23
Figure 28 Exercise 1 Report ccopt clock trees	24
Figure 29 Exercise 1 Nano Route	25
Figure 30 Exercise 1 Setup time PostRoute	26
Figure 31 Exercise 1 Hold time PostRoute	26
Figure 32 Exercise 1 Optimization PostRoute.....	26
Figure 33 Exercise 1 Verify DRC	27
Figure 34 Exercise 1 Results of DRC.....	28
Figure 35 Exercise 1 Verify Connectivity and Results	28
Figure 36 Exercise 1 Metal Fill Setup/Add	29
Figure 37 Exercise 1 Metal Fill Results	29
Figure 38 Exercise 2 Set Mode as High Power Effort.....	30
Figure 39 Exercise2 Nano Route	32
Figure 40 Exercise2 PostRoute SetupTime	33
Figure 41 Exercise2 PostRoute HoldTime	33
Figure 42 Exercise2 Physical Output	34
Figure 43 Exercise 3 85% Usage of CPU	35
Figure 44 Exercise3 Results of 85% usage CPU	36
Figure 45 Exercise2 Nano Route	39
Figure 46 Exercise 8 Before Pads	53
Figure 47 Exercise 8 After Pads.....	54

Εισαγωγή

Στην παρούσα εργασία θα υλοποιήσουμε την σύνθεση (logic synthesis), τη χωροθέτηση (floorplanning), και την τοποθέτηση και δρομολόγηση (placement & routing) χρησιμοποιώντας τα εργαλεία της Cadence.

Άσκηση 1

Στην αρχή πρέπει να εισαγάγουμε τα κατάλληλα paths για τις βιβλιοθήκες, Verilog folder, script folder με τις εξής εντολές:

```
set_db init_lib_search_path
/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsclib045
/

#setup script path
set_db script_search_path /mnt/scratch_b/users/n/ngiannop/Desktop/VLSI_ASIC22/script/

#setup hdl path
set_db init_hdl_search_path /mnt/scratch_b/users/n/ngiannop/Desktop/VLSI_ASIC22/verilog/

#Setup timing library
set_db library
/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsclib045
/timing/fast_vdd1v0_basicCells.lib
#end

#Setup lef library
set_db lef_library
{/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsclib045
/lef/gsclib045_tech.lef
/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsclib045
/lef/gsclib045_macro.lef}
#end

#Setup qrc library
read_qrc
/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsclib045
/qrc/qx/gpdk045.tch
#end

#setup hdl file
```

```
read_hdl picorv32.v
```

Πρωτού κάνω elaborate εφαρμοσα clock gating με τις εξής εντολές:

1. set_db lp_insert_clock_gating true
2. elaborate picorv32
3. syn_gen

και εκτελώντας την εντολή report_clock_gating βλέπουμε τα αποτελέσμα της εντολής αυτής.

```
@genus:root: 14> report_clock_gating
=====
Generated by:      Genus(TM) Synthesis Solution 19.11-s087_1
Generated on:      Oct 28 2022  10:38:21 pm
Module:            picorv32
Technology libraries: fast_vdd1v0 1.0
                   physical_cells
Operating conditions: PVT_1P1V_0C
Interconnect mode:  global
Area mode:         physical library
=====
Summary
-----
Info : ACTP-0001 [ACTPInfo] Activity propagation started for stim#0
Info : ACTP-0009 [ACTPInfo] Activity Propagation Progress Report : 100%
Info : ACTP-0001 Activity propagation ended for stim#0
-----
Category          Number    %    Average Toggle Saving %
-----
RC Clock Gating Instances      58   100      81.87
Non-RC Clock Gating Instances    0    0       0.00
-----
RC Gated Flip-flops      1704   87      86.27
Non-RC Gated Flip-flops     0    0       0.00
-----
Total Ungated Flip-flops      257   13       -
Enable not found             238   93       -
Register bank width too small   19    7       -
-----
Total Flip-flops      1961  100       -
-----
58
```

Figure 1 Exercise 1 Report Clock Gating

Όπου είναι εμφανές ότι έχουμε μια αποταμίευση 82% από Gating Instances και 87% από τα Flip-Flops

Ποιό είναι το top-level module του κυκλώματός σας; Με ποιά εντολή μπορώ να το προσδιορίσω αυτό;

```
@genus:root: 11> elaborate picorv32
Info : Elaborating Design. [ELAB-1]
Info : Elaborating top-level block 'picorv32' from file '/mnt/scratch_b/users/n/ngiannop/Desktop/VLSI_ASIC22/verilog/picorv32.v'.
```

Figure 2 Exercise 1 Top level Module

Το top-level module είναι το picorv32 όπου αυτό προσδιορίζεται με την σειρά που του δίνονται τα ανάλογα αρχεία Verilog από το βαθύτερο προς το υψηλότερο επίπεδο άρα αν έχουμε N αρχεία Verilog και το 1 είναι το βαθύτερο το 2 είναι το υψηλότερο επίπεδου από αυτό του 1 τότε το N είναι το υψηλότερο επίπεδο της ιεραρχίας αυτής.

Μετά το τέλος του elaborate κάνουμε check_design για να δούμε αν υπάρχουν σφάλματα στην περιγραφή της RTL όπου φαίνεται ότι δεν υπάρχει κανένα.

```
@genus:root: 16> check_timing_intent
=====
Generated by:      Genus(TM) Synthesis Solution 19.11-s087_1
Generated on:      Nov 05 2022  10:42:48 am
Module:           picorv32
Technology libraries: fast_vddlv0 1.0
                  physical_cells
Operating conditions: PVT_1P1V_0C
Interconnect mode:  global
Area mode:        physical library
=====

Lint summary
Unconnected/logic driven clocks                      0
Sequential data pins driven by a clock signal         0
Sequential clock pins without clock waveform          0
Sequential clock pins with multiple clock waveforms  0
Generated clocks without clock waveform              0
Generated clocks with incompatible options            0
Generated clocks with multi-master clock             0
Paths constrained with different clocks               0
Loop-breaking cells for combinational feedback       0
Nets with multiple drivers                           0
Timing exceptions with no effect                     0
Suspicious multi_cycle exceptions                   0
Pins/ports with conflicting case constants           0
Inputs without clocked external delays               0
Outputs without clocked external delays              0
Inputs without external driver/transition            0
Outputs without external load                        0
Exceptions with invalid timing start-/endpoints      0

Total: 0

@genus:root: 17> █
```

Figure 3 Exercise 1 Check Timing Intent

Μετά το elaborate & check_design προσθέτουμε την εντολή αυτή:

```
set_db / .use_scan_seqs_for_non_dft false
```

για να μην γίνει χρήση του scan Flip-Flop. Επιπλέον κάνουμε χρήση των 3^{ων} εντολών:

1. syn_generic
2. syn_map
3. syn_opt

όπου μετά το πέρας των εντολών αυτών εκτελούμε τα εξής report info:

1. report_area
2. report_gate
3. report_timing
4. report_power

Και προκύπτουν τα αποτελέσματα

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
8686	39052.744	+6071	1.538e ⁻⁰³

Καθώς το leakage power αποτελεί το 0.12% της συνολικής κατανάλωσης ρεύματος το οποίο είναι ένα πολύ καλό ποσοστό σε σχέση με την ολική κατανάλωση ρεύματος. Περαιτέρω μείωση αυτού του ρεύματος είναι ανέφικτη ή καθόλου κερδοφόρα.

Μετά την παραγωγή αρχείων από το Genus για το εργαλείο Innonus ανοίγοντας το με τα κατάλληλα path όπως αναφέρεται και στο εγχειρίδιο. Ξεκινάμε με το placement βάζοντας ως 75% χρήση του πυρήνα και απόσταση από E/E 15 μm .



Figure 4 Exercise 1 FloorPlan

Προκύπτει το εξής φυσικό σχήμα:

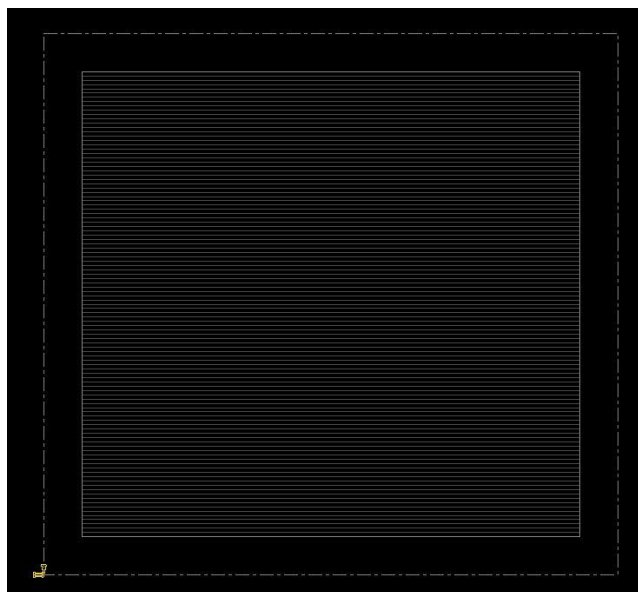


Figure 5 Exercise 1 Results of FloorPlan

Στη συνέχεια κατευθυνόμαστε στην τοποθέτηση του δικτύου διανομής ενέργειας ξεκινώντας με:

1. Ring
2. Stripe

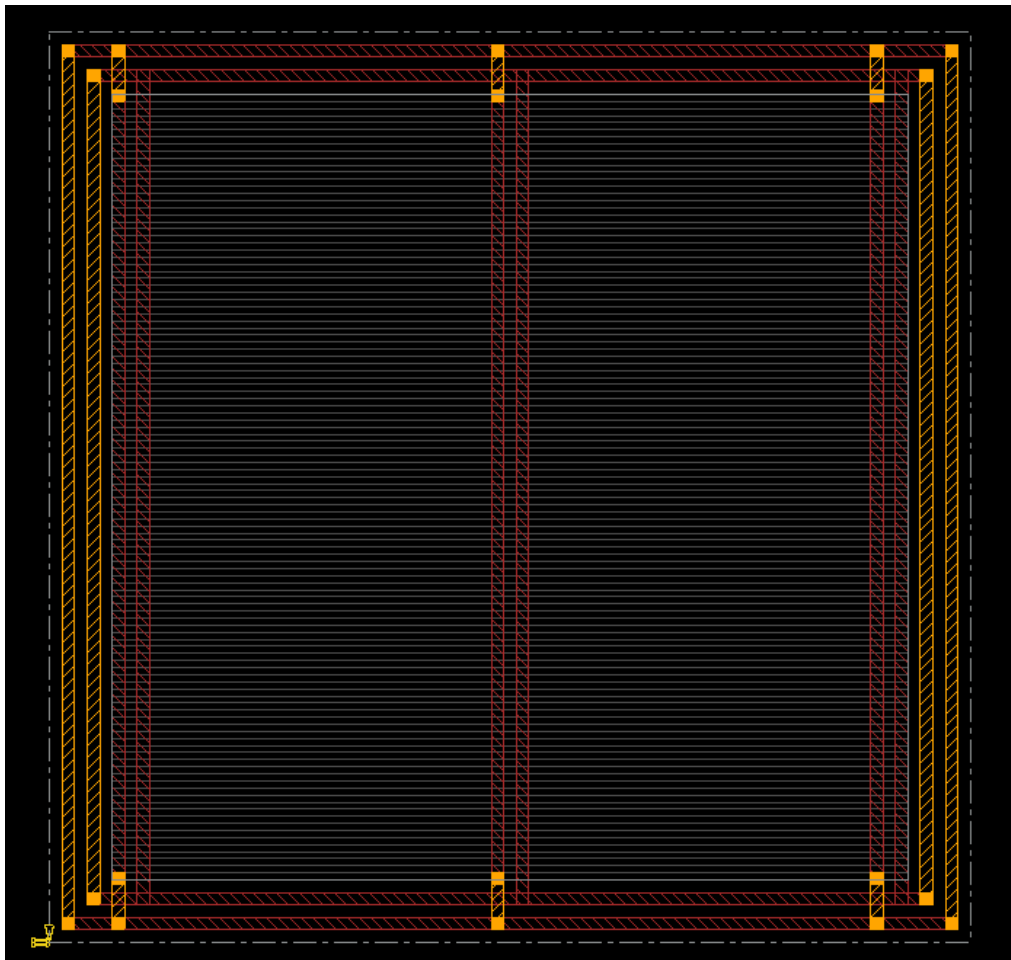
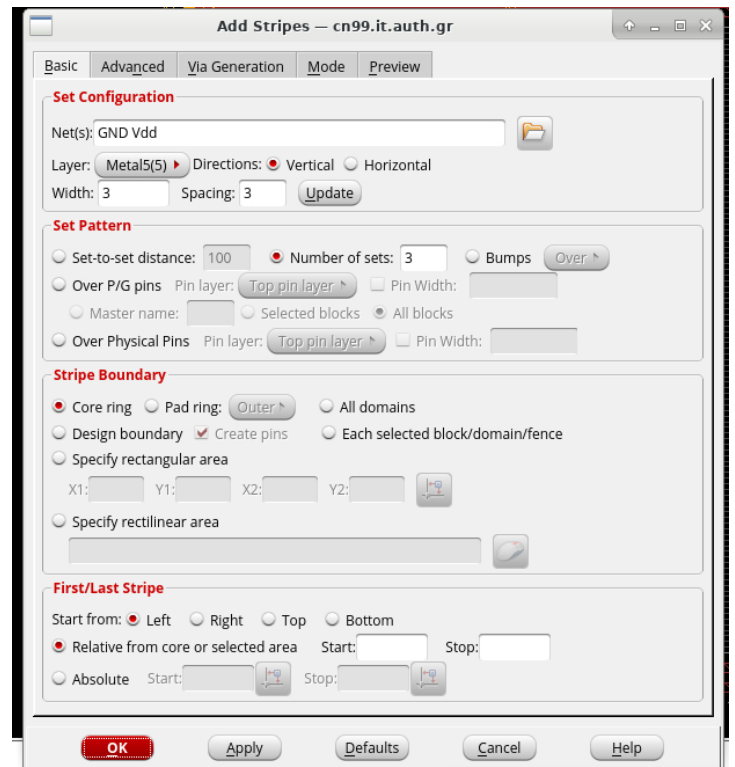


Figure 6 Exercise 1 Rings & Stripes and output

Στη συνέχεια δημιουργούμε τα follow pins με τις εξής εντολές:

Για το Vdd:

- `globalNetConnect Vdd -type pgpin -pin VDD -inst *`
- `globalNetConnect Vdd -type tiehi -instanceBasename *`

Για το GND:

- `globalNetConnect GND -type pgpin -pin VSS -inst *`
- `globalNetConnect GND -type tielo -instanceBasename *`

Επιπλέον πρέπει να δημιουργήσουμε τους ακροδέκτες τροφοδοσίας και γείωσης στο φυσικό επίπεδο καθώς δεν υπάρχουν από πριν στο netlist με τις εξής εντολές:

- `createPGPin VDD -net Vdd -geom Metal6 20 0 20.8 0.08`
- `createPGPin GND -net GND -geom Metal7 10 0 10.8 0.08`

Και προκύπτει το εξής αποτέλεσμα:

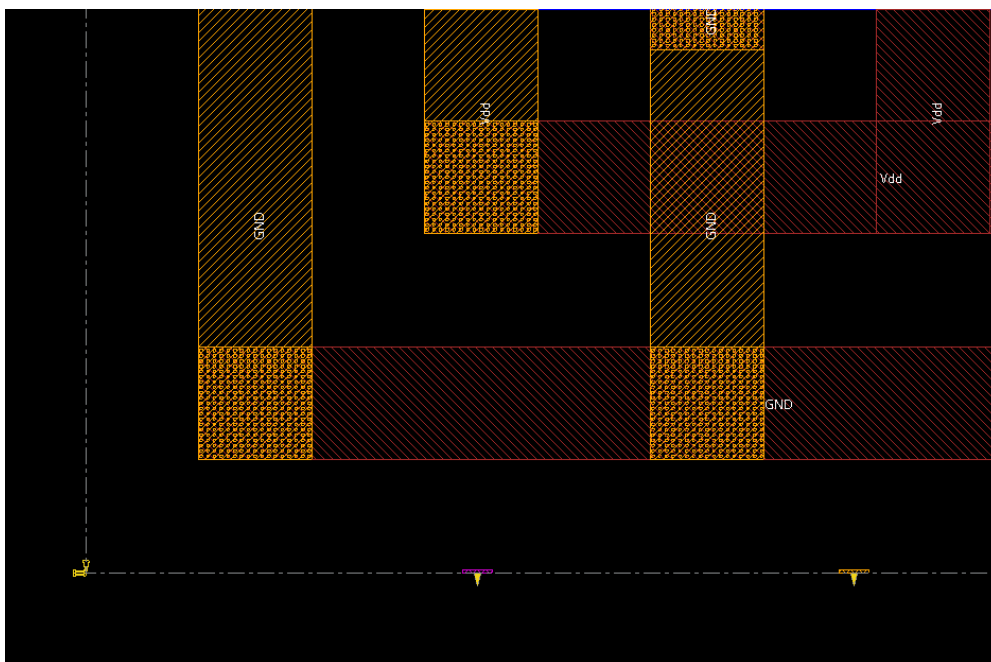
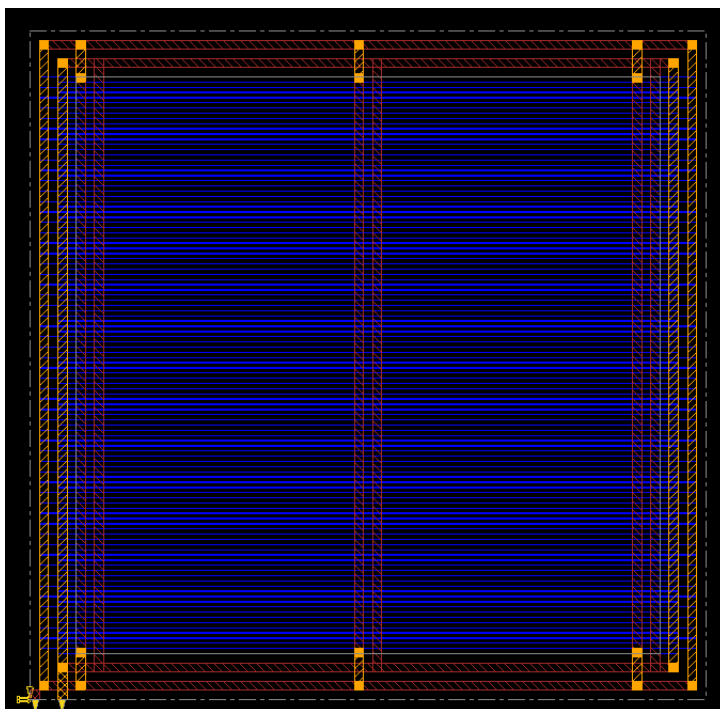


Figure 7 Exercise 1 Pins Vdd/GND

Και για να συνδέσουμε τα δυο επίπεδα μετάλλου (5,6) κάνουμε τα εξής:

- `editPowerVia -add_vias 1 -top_layer Metal6 -area {10 0 10.8 0.08} -bottom_layer Metal5 (για το GND)`
- `editPowerVia -add_vias 1 -top_layer Metal6 -area {20 0 20.8 0.08} -bottom_layer Metal5 (για Vdd)`

Έπειτα κάνοντας τις κατάλληλες επιλογές από το μενού Router->Special Route.



Καταλήγουμε στο ανάλογο αποτέλεσμα.

Figure 8 Exercise 1 SRoute & results

Πρωτού προχωρήσουμε στο Rail Analysis γράφουμε στην γραμμή εντολών του Ilnonus:

- `place_opt_design`

Όπου θα τοποθετηθούν οι κατάλληλοι ακροδέκτες E/E του κυκλώματος, και προκύπτει το εξής:

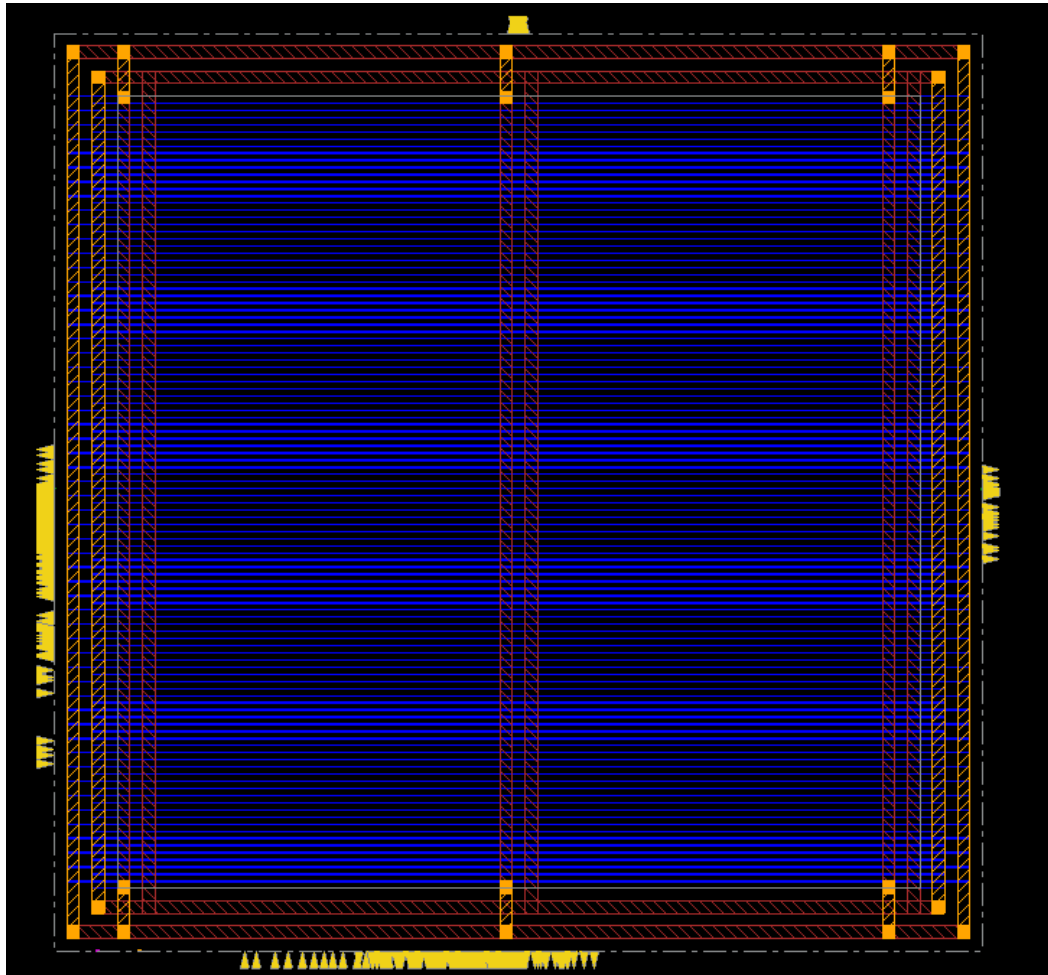


Figure 9 Exercise 1 Place I/O results

Επιπλέον θα ελέγξουμε αν η τοποθέτηση έγινε σωστά πάμε Place → Check Placement και προκύπτει το εξής:

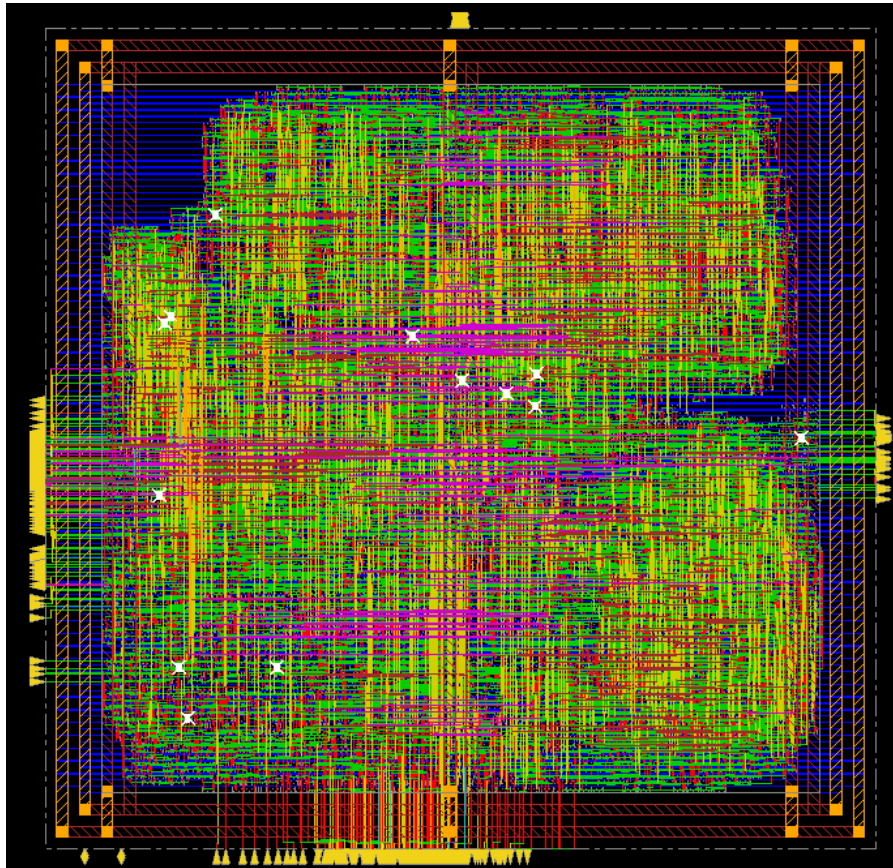


Figure 10 Exercise 1 Check Placement

Καθώς κάνουμε και έλεγχο/διόρθωση με Place → Refine Placement → Preserve Routing Check → OK.

Επιπλέον για να αποφύγουμε την συμφόρηση που μπορεί να δημιουργηθεί σε φυσικό σχέδιο κάνουμε Route → Early Global Route.

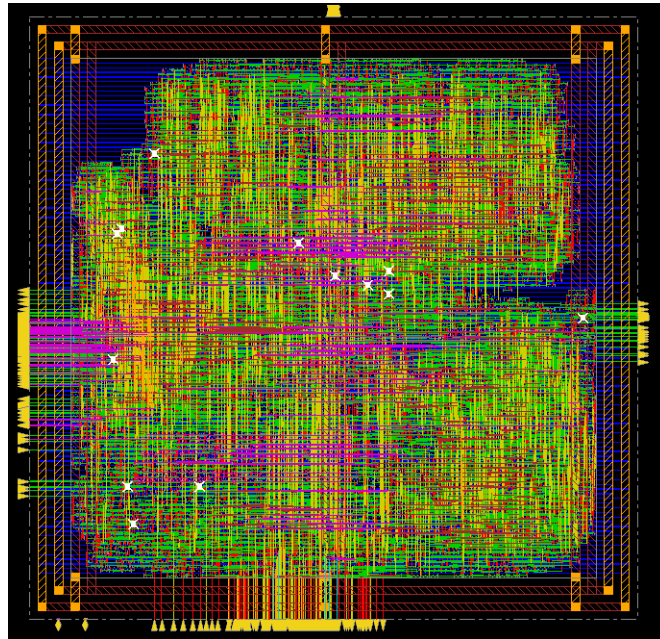


Figure 11 Exercise 1 Early Global Route

Τώρα θα κάνουμε Timing Analysis:

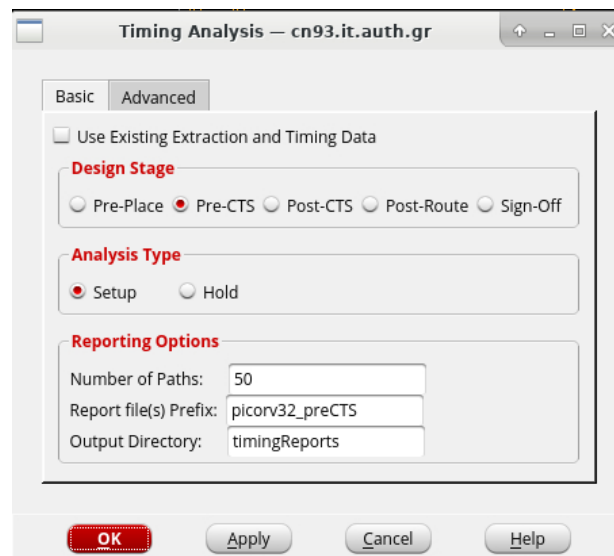


Figure 12 Exercise 1 Timing Analysis with Pre-CTS and Setup Time

Και προκύπτει:

timeDesign Summary				
Setup views included: default_emulate_view				
Setup mode	all	reg2reg	reg2cgate	default
WNS (ns):	6.330	6.587	8.259	6.330
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0
All Paths:	2342	2032	59	722

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	234 (234)	-0.204	234 (234)
max_tran	1 (1)	-0.236	1 (1)
max_fanout	0 (0)	0	0 (0)
max_length	0 (0)	0	0 (0)

Figure 13 Exercise 1 Results setup time

Όπου το $\text{WorstNegativeSlack}_{\text{all}} = 6.330$ και $\text{TotalNegativeSlack} = 0$ το οποίο είναι θετικό και άρα ικανοποιούμε τις απαιτήσεις του κυκλώματος.

Για το hold time:

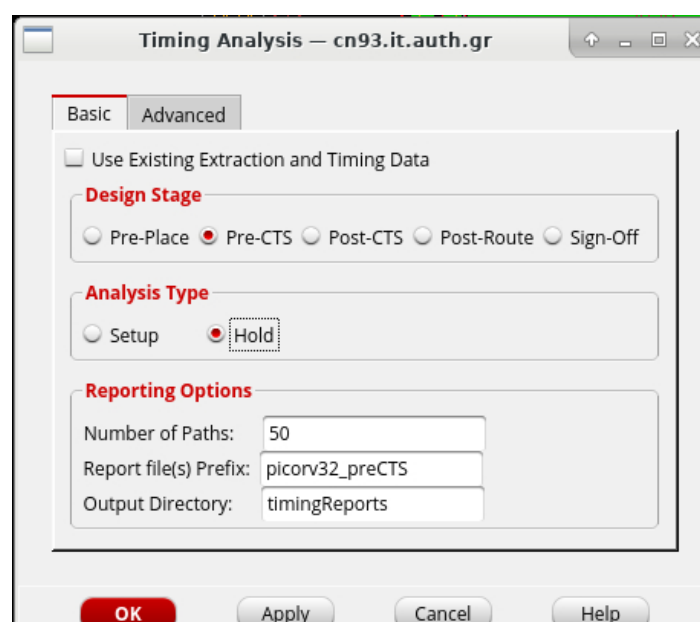


Figure 14 Exercise 1 Timing Analysis with Pre-CTS and Hold Time

Και προκύπτει:

```
-----
timeDesign Summary
-----

Hold views included:
default_emulate_view

+-----+-----+-----+-----+-----+
| Hold mode | all | reg2reg | reg2cgate | default |
+-----+-----+-----+-----+-----+
| WNS (ns): | 0.007 | 0.007 | 0.043 | 0.342 |
| TNS (ns): | 0.000 | 0.000 | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 | 0 | 0 |
| All Paths: | 2342 | 2032 | 59 | 722 |
+-----+-----+-----+-----+-----+
```

Figure 15 Exercise 1 Results hold time

$WorstNegativeSlack_{all} = 0.007$ και $TotalNegativeSlack = 0.000$ στην περίπτωση του Hold είμαστε οριακά καλυμμένοι με περαιτέρω πρόσθεση καθυστέρησης για διάφορους λόγους (π.χ πρόσθετο υλικό στην περιγραφή της RTL) θα υπήρχε περίπτωση να ήταν αρνητικό και να μην μας καλύπτει στις απαιτήσεις του κυκλώματος έτσι θα έπρεπε να γίνει αλλαγή της RTL περιγραφής.

Τώρα θα κάνουμε Power Rail Analysis για να βεβαιωθούμε ότι το κύκλωμα τροφοδοτείται σε όλα τα σημεία του και δεν υπάρχει περίπτωση να έχουμε χαμηλότερη τάση από την ελάχιστη δυνατή.

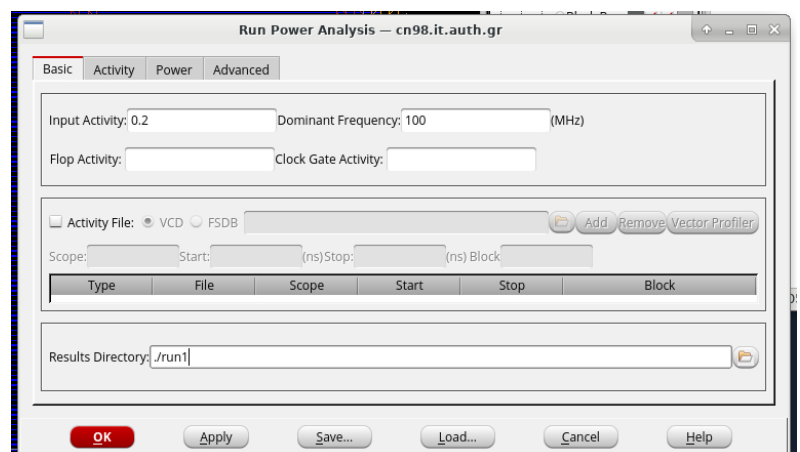
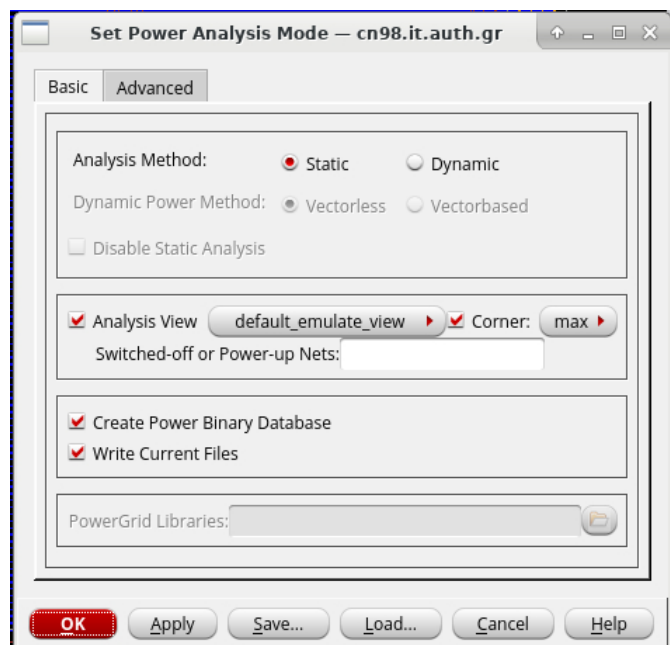


Figure 16 Exercise 1 Set Power Analysis Mode step 1

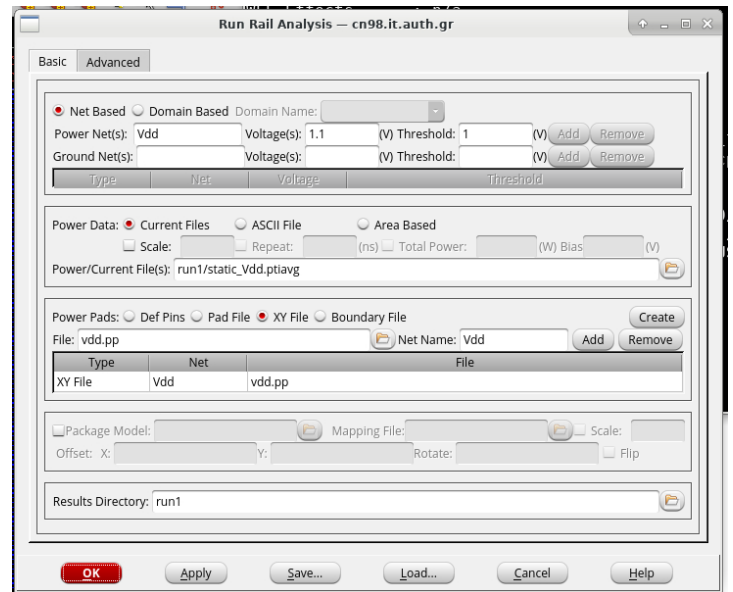
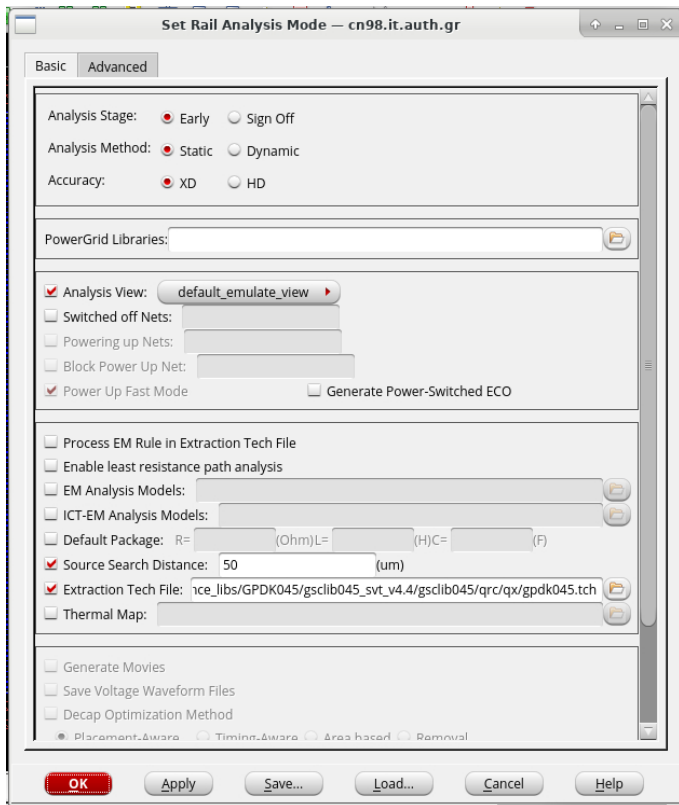


Figure 17 Exercise 1 Set Power Analysis Mode step 2

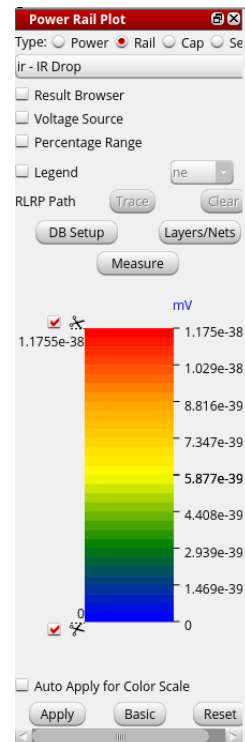
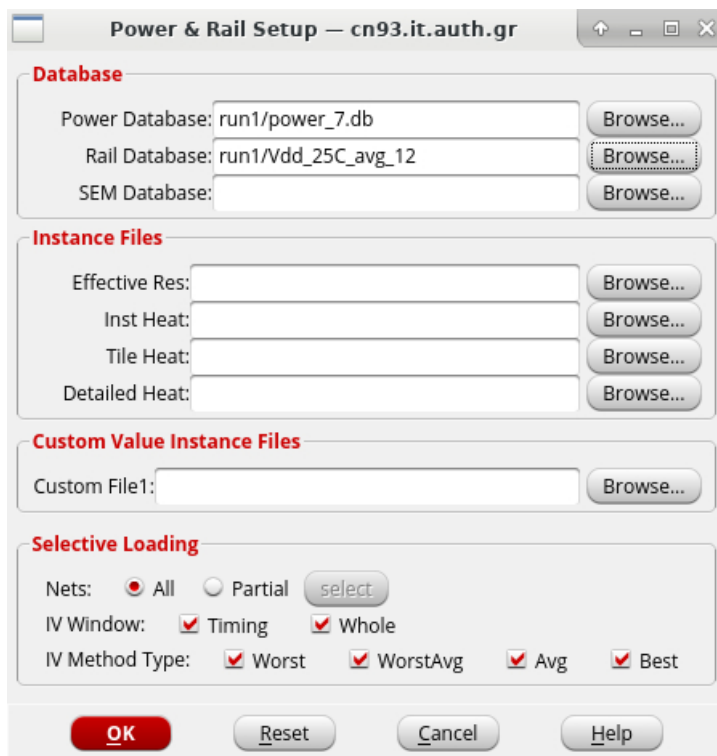


Figure 18 Exercise 1 Set Power Analysis Mode step 3 & Power Rail Plot

Όπου τα αποτελέσματα που προκύπτουν είναι αρκετά ικανοποιητικά καθώς έχουμε απώλεια της τάξης του $\sim 10^{-39}$

Στη συνέχεια επιλέγουμε (ήταν ήδη προεπιλογή) ως Timing Effort
 → High και Power Effort → None και προκύπτουν τα εξής:

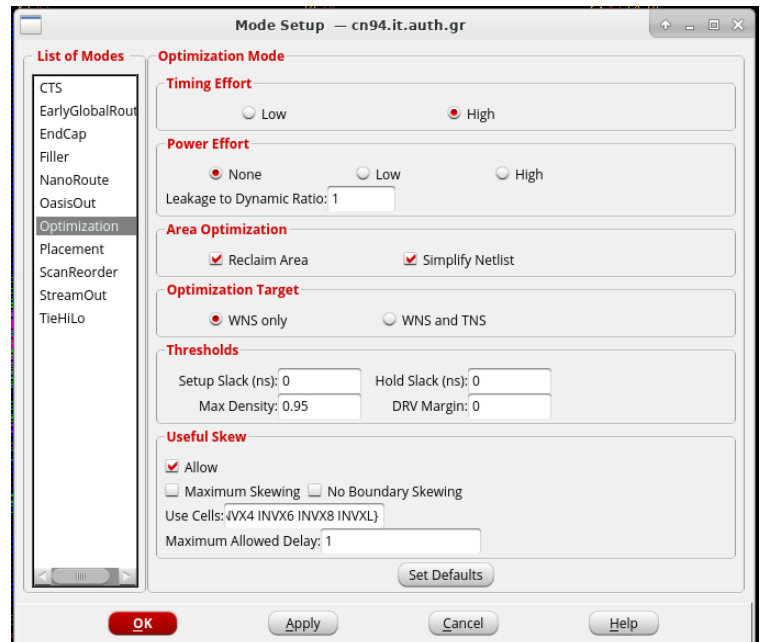


Figure 19 Exercise 1 Mode Setup Optimization

Slack	Setup time: WNS = 6.222	Hold Time: WNS = 0.008
Power	Internal Power Switching Power Leakage Power	% 26.58 (0.761) 73.34(2.1) 0.0645(0.0018)
Area	27245.088	

Επιπλέον θα επιλέξουμε ως Early Global Routing με εύρος δρομολόγησης από όλα τα μέταλλα.

1. Place → Check Placement
2. Place → Refine Placement → Preserve Routing Check → OK
3. Route → Early Global Route

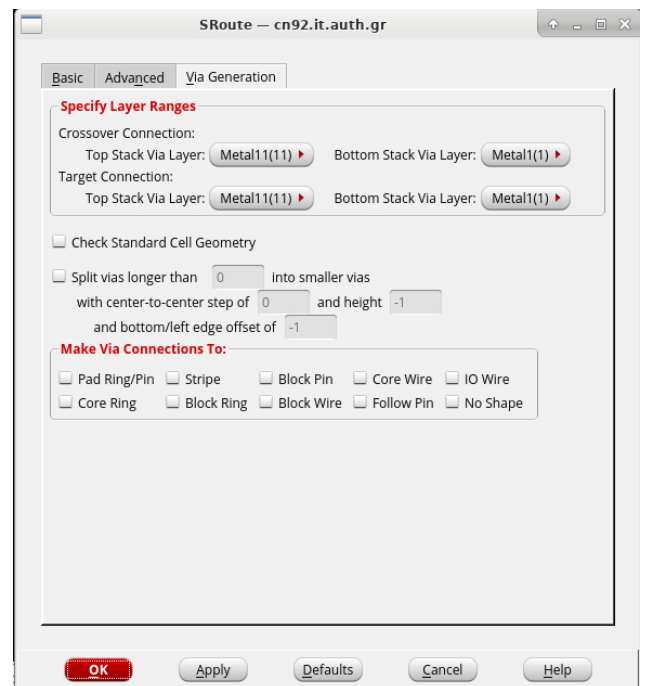


Figure 20 Exercise 1 SRoute All levels of metals

Και εμφανίζονται
περισσότερες συμφορήσεις.

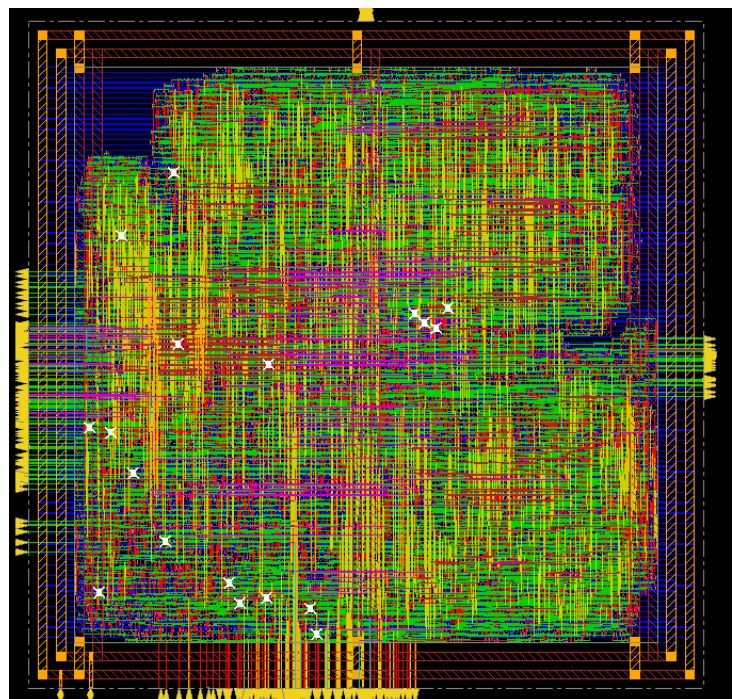
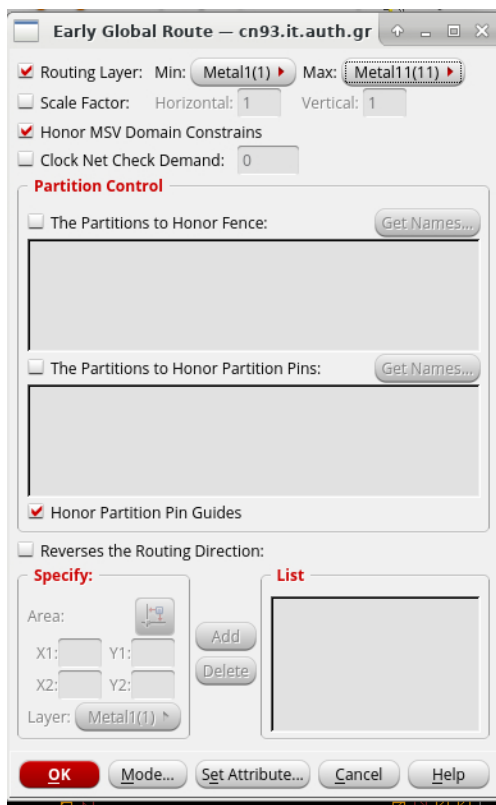


Figure 21 Exercise 1 Early Global Route all metals & results

Τώρα θα κάνουμε το ίδιο αλλά για τα μέταλλα M2 έως M10.

1. Place → Check Placement
2. Place → Refine Placement → Preserve Routing Check → OK
3. Route → Early Global Route

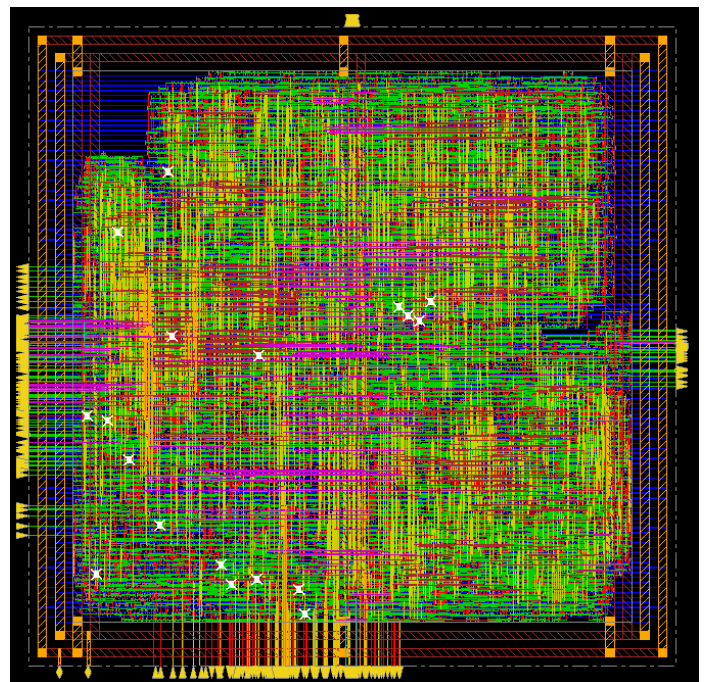
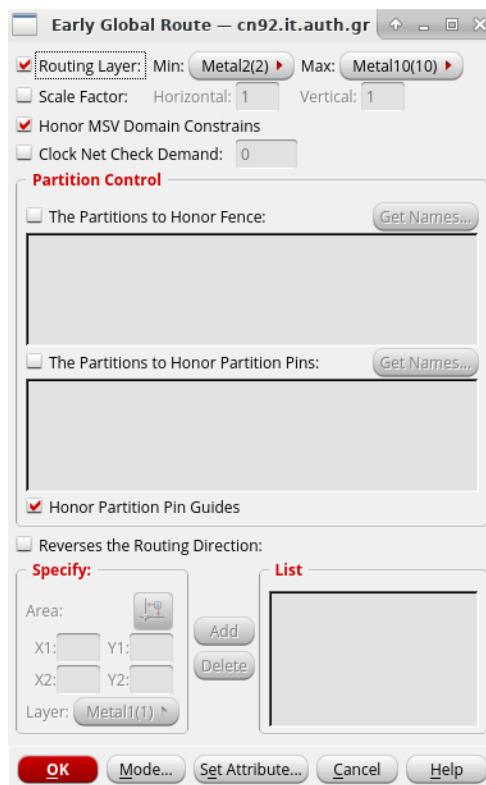
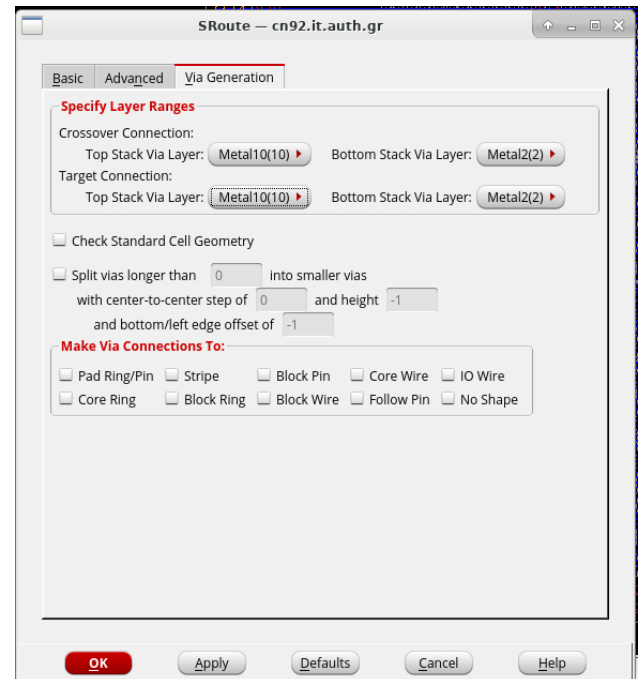


Figure 22 Exercise 1 SRoute Metal2-Metal10 levels of metals & Results

Όπου συνεχίζουν να υπάρχουν οι ίδιες συμφορήσεις.

Στη συνέχεια θα εκτελέσουμε την σύνθεση του δέντρου ρολογιού πηγαίνοντας:

- Edit → Create Non Default Rules

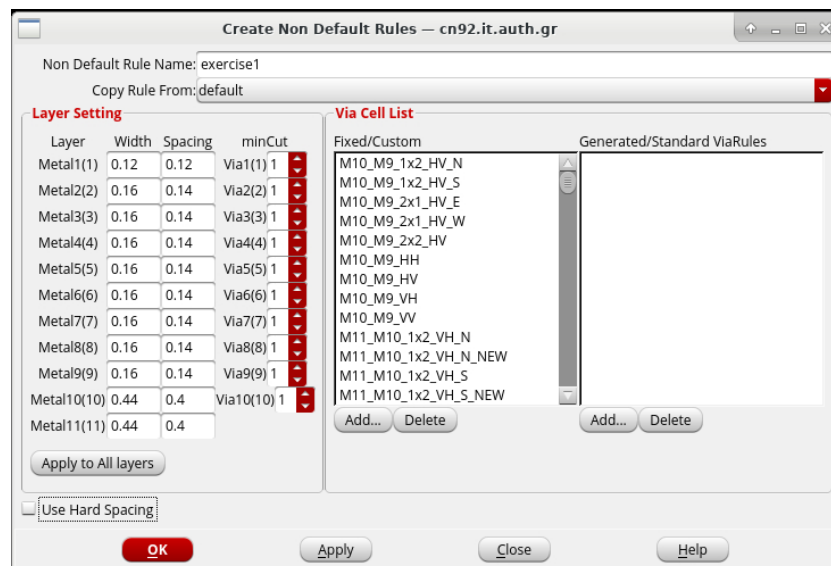


Figure 23 Exercise 1 Create Non Default Rules

Όπου άλλαξα τις ελάχιστες στις διπλάσιες τιμές Width, Spacing.

Στη συνέχεια θα χρησιμοποιήσουμε την γραμμή εντολών με τις εξής εντολές

- create_route_type -top_preferred_layer 6 -bottom_preferred_layer 4 -non_default_rule exercise1 -name exercise1_clock_tree
- set_ccopt_property target_skew 0.1
- set_ccopt_property target_max_trans 0.15

Και στη συνέχεια κάνουμε:

- create_ccopt_clock_tree_spec -file tree_step_14_exercise1.spec

Και για να τοποθετήσουμε το δέντρο διανομής ρολογιού:

- ccopt_design

Μετά το πέρας της εντολής αυτής πάμε και κάνουμε:

- Clock → Clock Tree Debugger → OK → View → Enable clock path browser

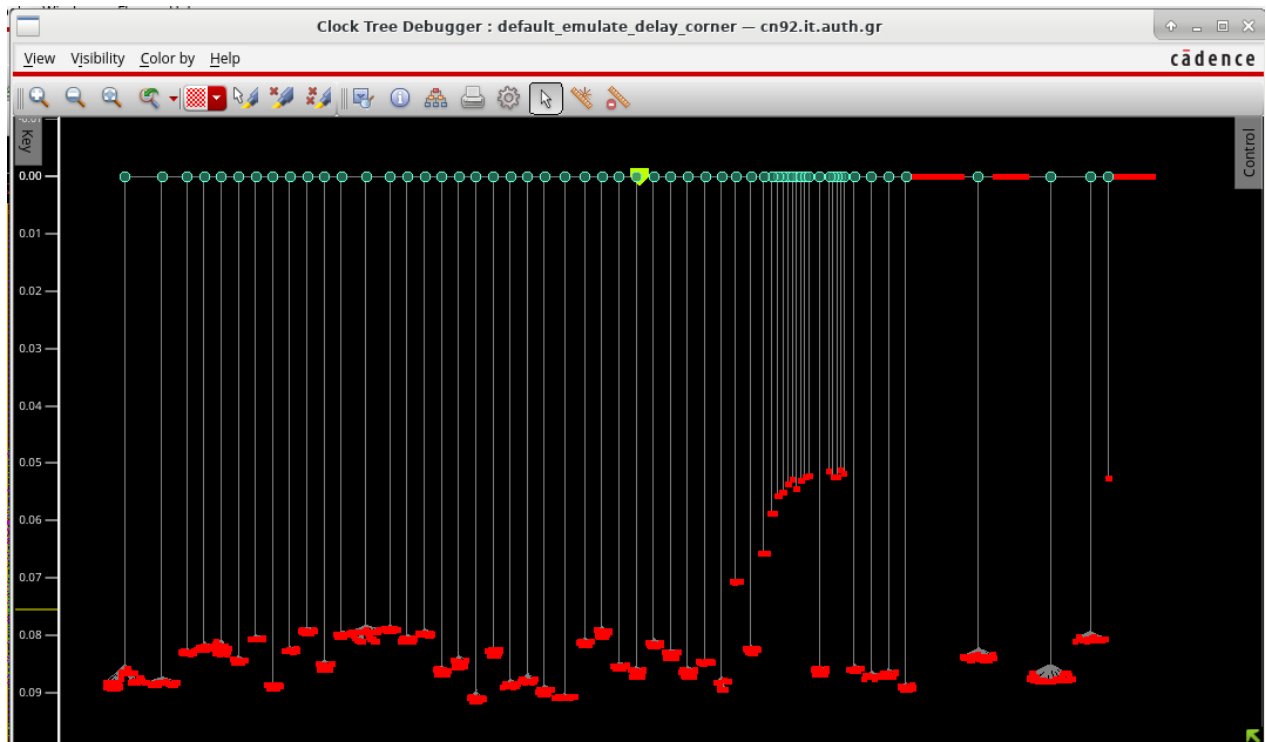


Figure 24 Exercise 1 Clock Tree Debugger

The image shows the 'Clock Path Browser' window from Cadence. The title bar reads 'Clock Path Browser — cn92.it.auth.gr'. The window displays a table with the following columns: Analysis View, Skew Group, Skew, Min Delay, Max Delay, Min Pin, MinPath Level, Max Pin, and MaxPath Level. The table contains five rows of data, all with a skew of 0.091 and a max delay of 0.091. A 'Close' button is located at the bottom of the window.

Analysis View	Skew Group	Skew	Min Delay	Max Delay	Min Pin	MinPath Level	Max Pin	MaxPath Level
...late_delay_corner:hold.early	...emulate_constraint_mode	0.091	0.000	0.091	...rdsz_reg[1]/CK	2	...s_reg[11][15]/CK	3
...ulate_delay_corner:hold.late	...emulate_constraint_mode	0.091	0.000	0.091	...rdsz_reg[1]/CK	2	...s_reg[11][15]/CK	3
...ate_delay_corner:setup.early	...emulate_constraint_mode	0.091	0.000	0.091	...rdsz_reg[1]/CK	2	...s_reg[11][15]/CK	3
...late_delay_corner:setup.late	...emulate_constraint_mode	0.091	0.000	0.091	...rdsz_reg[1]/CK	2	...s_reg[11][15]/CK	3

Figure 25 Exercise 1 Example of Clock Tree Debugger

Ακολουθώντας την βελτιστοποίηση για να εξάγουμε τα αποτελέσματα χρησιμοποιούμε:

- `optDesign -postCTS`

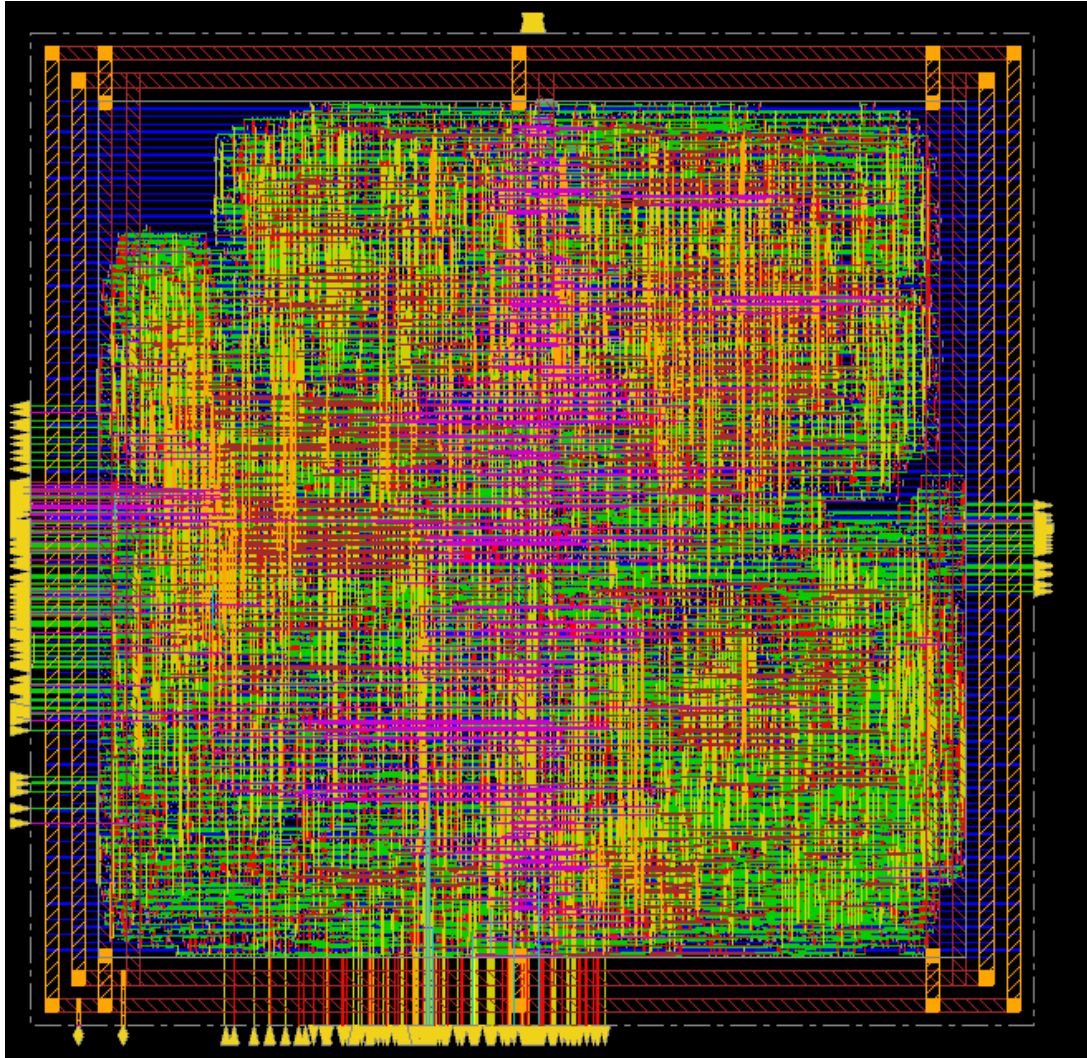


Figure 26 Exercise 1 Optimazation PostCTS

Έτσι μπορούμε να εξάγουμε τα αποτελέσματα για την επιφάνεια, Slack και κατανάλωση ενέργειας.

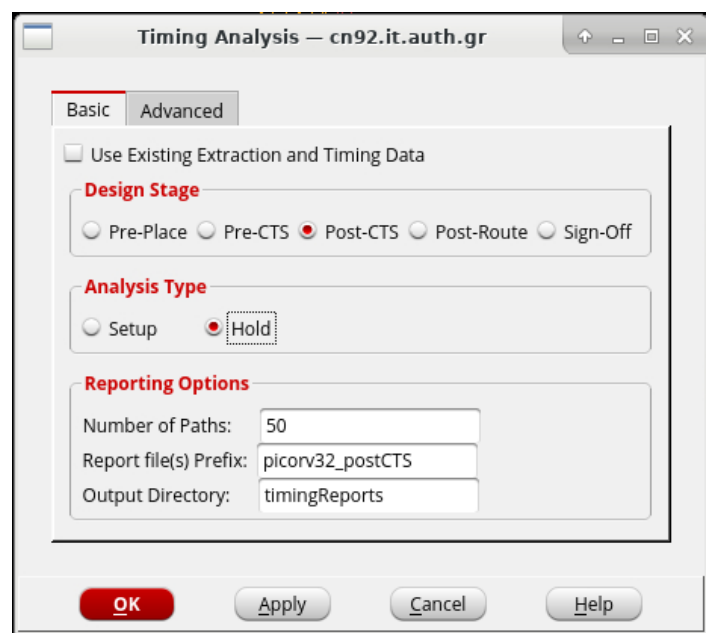
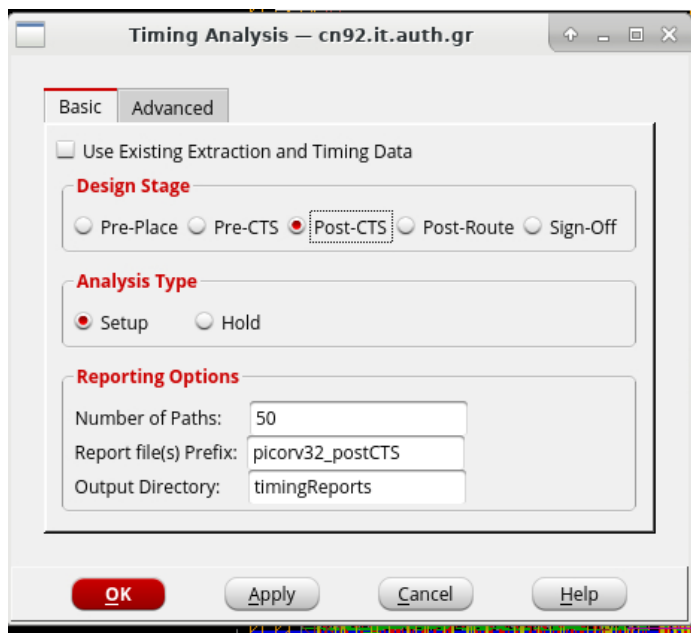


Figure 27 Exercise 1 Timing Analysis Setup/Hold Time Post-CTS

- report_area
- report_power

Slack	Setup time: WNS = 6.179	Hold Time: WNS= -0.032
Power	Internal Power Switching Power Leakage Power	% 26.7645 (0.777) 73.1709 (2.12) 0.0646 (0.0018)
Area	27469.782	

Όπως φαίνεται στο Hold-time είμαστε κοντά στο μηδέν οπότε μπορούμε να συνεχίσουμε παρακάτω χωρίς κάποια τροποποίηση.

Number of Buffers	0
Number of Skew Group	8
Maximum depth of clock tree	1
Minimum depth of clock tree	0
Routing length for the trunk	0
Routing length for the leafes	11129.230

Για να προκύψουν τα αποτελέσματα αυτά κάνουμε:

- report_ccopt_clock_trees
- report_ccopt_skew_groups

Το εργαλείο μας δείχνει ότι τα δέντρα ρολογιού να είναι unbuffered και έτσι δεν τα αναγνωρίζει.

Τα οποία ικανοποιούν τις συνθήκες στρέβλωση & μέγιστο ρυθμό μετάβασης όπως προκύπτει από την εντολή report_ccopt_clock_trees όπου φαίνεται στην εικόνα:

Clock DAG primary half-corner transition distribution:									
Net Type	Target	Count	Average	Std. Dev.	Min	Max	Distribution	Over Target	
Leaf	0.150	60	0.054	0.025	0.004	0.083	{60 <= 0.090ns, 0 <= 0.120ns, 0 <= 0.135ns, 0 <= 0.142ns, 0 <= 0.150ns}	-	-

Figure 28 Exercise 1 Report ccopt clock trees

Στη συνέχεια μετά την σύνθεση του δέντρου διανομής κάνουμε δρομολόγηση με:

- Route → NanoRoute → Route
- Route → NanoRoute → Route → Mode

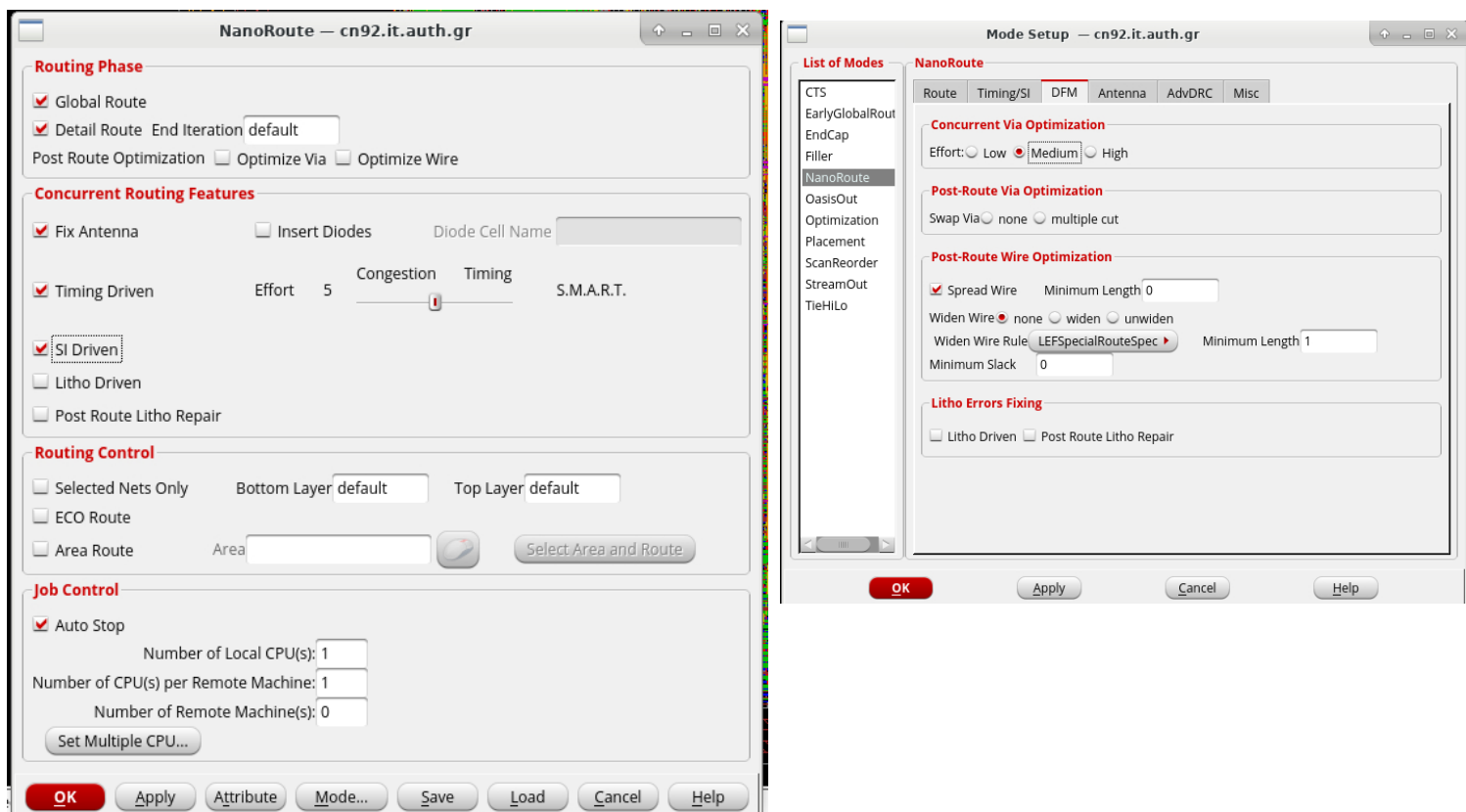


Figure 29 Exercise 1 Nano Route

Και στη συνέχεια κάνουμε:

- setAnalysisMode -analysisType onChipVariation
- optDesign -postRoute

ώστε να φύγει η κάθε συμφόρηση που δημιουργείται.

Για να ελέγξουμε τα hold/setup κάνουμε τα εξής:

1. Place → Check Placement
2. Place → Refine Placement → Preserve Routing Check → OK
3. Timing → Report Timing → Post-Route & Setup or Hold

Και προκύπτει το εξής αποτέλεσμα:

```
Setup views included:
default_emulate_view
```

Setup mode	all	reg2reg	reg2cgate	default
WNS (ns):	6.160	6.459	8.219	6.160
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0
All Paths:	2342	2032	59	722

Figure 30 Exercise 1 Setup time PostRoute

Όπου μας ικανοποιούν τα αποτέλεσμα για setup time το ίδιο κάνουμε και για hold time.

```
Hold views included:
default_emulate_view
```

Hold mode	all	reg2reg	reg2cgate	default
WNS (ns):	-0.036	-0.036	0.037	0.278
TNS (ns):	-0.663	-0.663	0.000	0.000
Violating Paths:	35	35	0	0
All Paths:	2342	2032	59	722

Density: 75.298%

Figure 31 Exercise 1 Hold time PostRoute

Για να φύγει κάθε περιορισμός εκτελούμε την παρακάτω εντολή στην γραμμή εντολών του Ilnovus:

- `optDesign -postRoute -setup -hold`

Setup mode	all	reg2reg	reg2cgate	default
WNS (ns):	6.158	6.459	8.216	6.158
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0
All Paths:	2342	2032	59	722

Hold mode	all	reg2reg	reg2cgate	default
WNS (ns):	-0.001	-0.001	0.037	0.278
TNS (ns):	-0.001	-0.001	0.000	0.000
Violating Paths:	1	1	0	0
All Paths:	2342	2032	59	722

Figure 32 Exercise 1 Optimization PostRoute

Και έτσι βλέπουμε την βελτίωση η οποία έγινε και είναι πολύ κοντά στο 0.

Slack	Setup time: WNS = 6.158	Hold Time: WNS= -0.001
Power	Internal Power Switching Power Leakage Power	% 26.8168 (0.778) 73.1118 (2.12) 0.0648 (0.0018)
Area	27554.598	

Το τελευταίο βήμα είναι ο έλεγχος για την ύπαρξη παραβάσεων με την επιλογή:

- Verify → Verify DRC



Figure 33 Exercise 1 Verify DRC

```

*** Starting Verify DRC (MEM: 1779.8) ***

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 74.880 73.920} 1 of 9
VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {74.880 0.000 149.760 73.920} 2 of 9
VERIFY DRC ..... Sub-Area : 2 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {149.760 0.000 222.200 73.920} 3 of 9
VERIFY DRC ..... Sub-Area : 3 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {0.000 73.920 74.880 147.840} 4 of 9
VERIFY DRC ..... Sub-Area : 4 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {74.880 73.920 149.760 147.840} 5 of 9
VERIFY DRC ..... Sub-Area : 5 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {149.760 73.920 222.200 147.840} 6 of 9
VERIFY DRC ..... Sub-Area : 6 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {0.000 147.840 74.880 219.830} 7 of 9
VERIFY DRC ..... Sub-Area : 7 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {74.880 147.840 149.760 219.830} 8 of 9
VERIFY DRC ..... Sub-Area : 8 complete 0 Viols.
VERIFY DRC ..... Sub-Area: {149.760 147.840 222.200 219.830} 9 of 9
VERIFY DRC ..... Sub-Area : 9 complete 0 Viols.

Verification Complete : 0 Viols.

*** End Verify DRC (CPU: 0:00:05.4 ELAPSED TIME: 6.00 MEM: 0.0M) ***

```

Figure 34 Exercise 1 Results of DRC

Και για να ελέγξουμε αν τυχόν υπάρχουν ασύνδετοι ακροδέκτες κάνουμε το εξής:

- Verify → Verify Connectivity

```

VERIFY_CONNECTIVITY use new engine.

***** Start: VERIFY CONNECTIVITY *****
Start Time: Mon Nov 21 14:39:36 2022

Design Name: picorv32
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (222.2000, 219.8300)
Error Limit = 1000; Warning Limit = 50
Check all nets
**** 14:39:37 **** Processed 5000 nets.
**** 14:39:37 **** Processed 10000 nets.

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Mon Nov 21 14:39:37 2022
Time Elapsed: 0:00:01.0

***** End: VERIFY CONNECTIVITY *****
Verification Complete : 0 Viols. 0 Wrngs.
(CPU Time: 0:00:01.0 MEM: 0.000M)

```

Figure 35 Exercise 1 Verify Connectivity and Results

Στη συνέχεια θα κάνουμε το γέμισμα των κελιών.

- Route → Metal Fill → Setup/Add

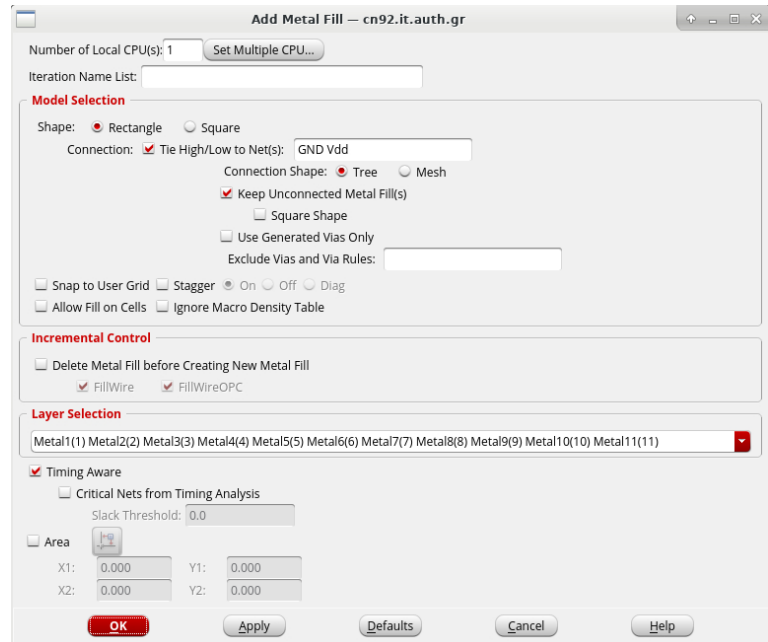
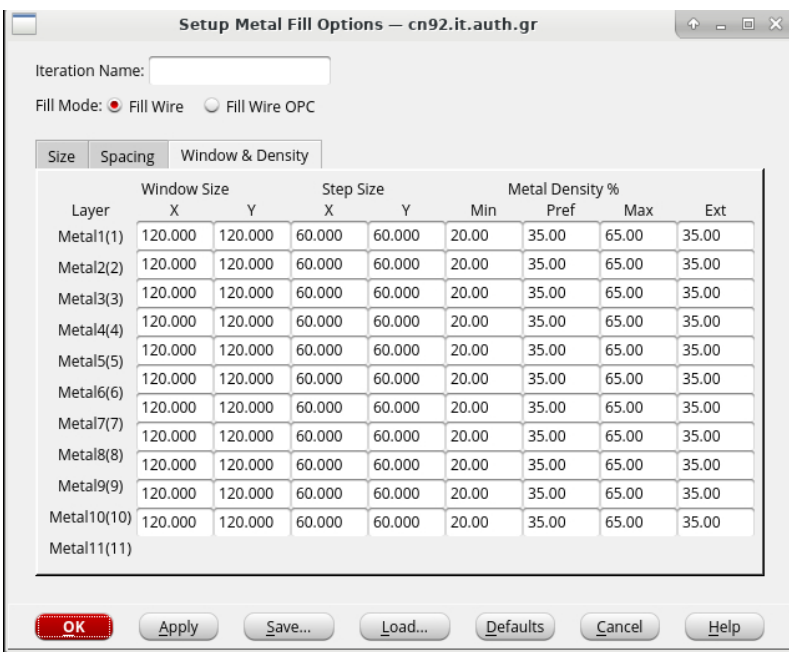
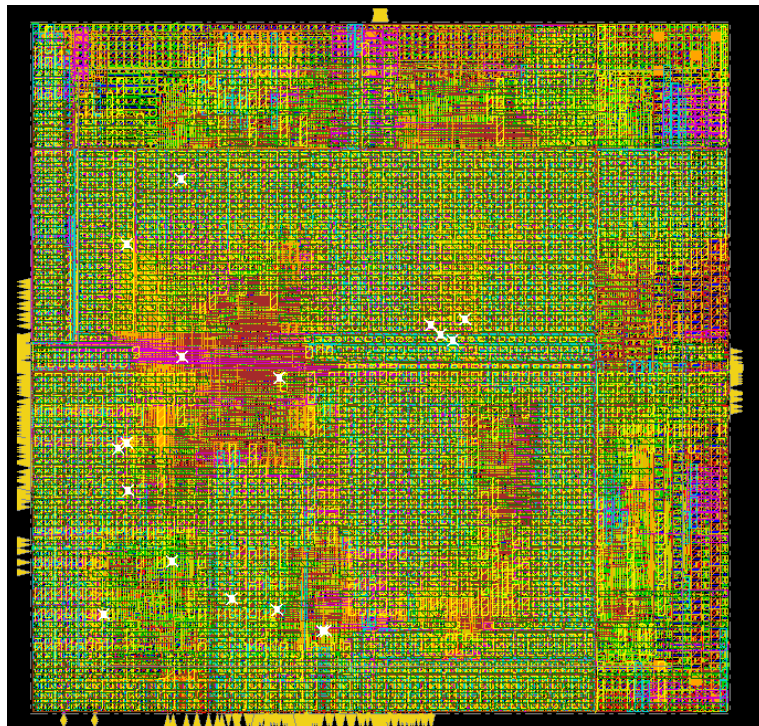


Figure 36 Exercise 1 Metal Fill Setup/Add

Και προκύπτει το εξής αποτέλεσμα:

Figure 37 Exercise 1 Metal Fill Results



Άσκηση 2

Αλλάζοντας τις επιλογές του Optimization:

- Tools → Set Mode → Optimization
- Στην γραμμή εντολών Innonus: place_opt_design

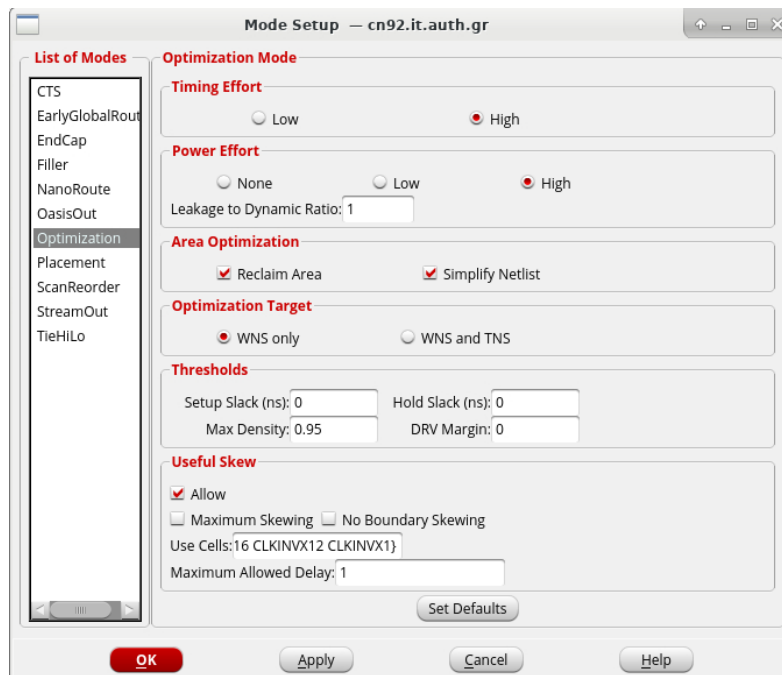


Figure 38 Exercise 2 Set Mode as High Power Effort

Στη συνέχεια εκτελούμε τα εξής:

1. Place → Check Placement
2. Place → Refine Placement → Preserve Routing Check → OK
3. optDesign

Προκύπτουν τα εξής:

Slack	Setup time: WNS = 6.304	Hold Time: WNS = 0.003
Power	Internal Power Switching Power Leakage Power	% 24.6161 (0.6898) 75.32(2.11) 0.0616(0.0017)
Area	27876.420	

Όπου υπάρχουν διαφορές στην ενέργεια καθώς μειώθηκε το leakage power αλλά αυξήθηκε περίπου ~2% το Switching Power καθώς και το Internal Power μειώθηκε σε ~2%. Επιπλέον αυξήθηκε και το Slack του Setup Time

Στη συνέχεια κάνουμε:

- Edit → Create Non Default Rules

Στη συνέχεια θα χρησιμοποιήσουμε την γραμμή εντολών με τις εξής εντολές:

- create_route_type -top_preferred_layer 6 -bottom_preferred_layer 4 -non_default_rule exercise2_2 -name exercise2_8clock_tree
- set_ccopt_property target_skew 0.1
- set_ccopt_property target_max_trans 0.15

Και στη συνέχεια κάνουμε:

- delete_ccopt_clock_tree_spec
- create_ccopt_clock_tree_spec -file tree_step_14_exercise2.spec

Και για να τοποθετήσουμε το δέντρο διανομής ρολογιού:

- ccopt_design
- report_ccopt_clock_trees
- report_ccopt_skew_groups

Number of Buffers	0
Number of Skew Group	8
Maximum depth of clock tree	1
Minimum depth of clock tree	1
Routing length for the trunk	0
Routing length for the leafes	7675.710

Όπως και πριν το εργαλείο μας δείχνει ότι τα δέντρα ρολογιού να είναι unbuffered και έτσι δεν τα αναγνωρίζει.

Στη συνέχεια θα εκτελέσουμε δρομολόγηση επιλέγοντας:

- Route → NanoRoute → Route
- Route → NanoRoute → Route → Mode

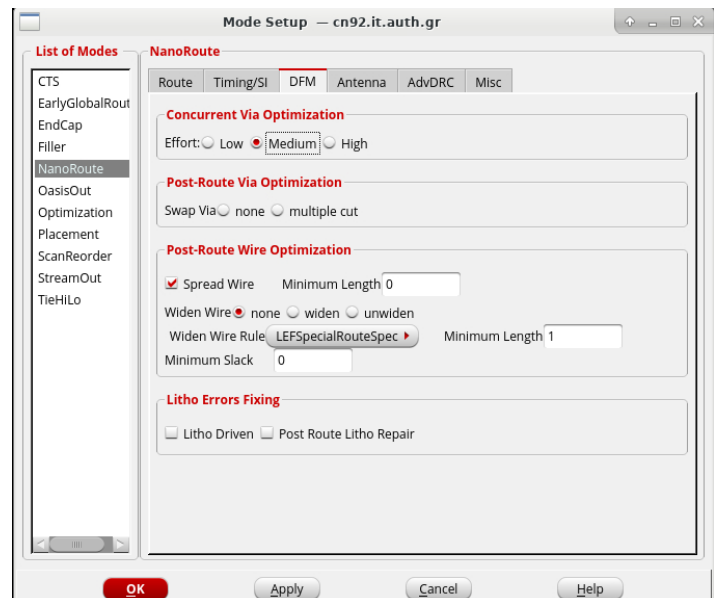
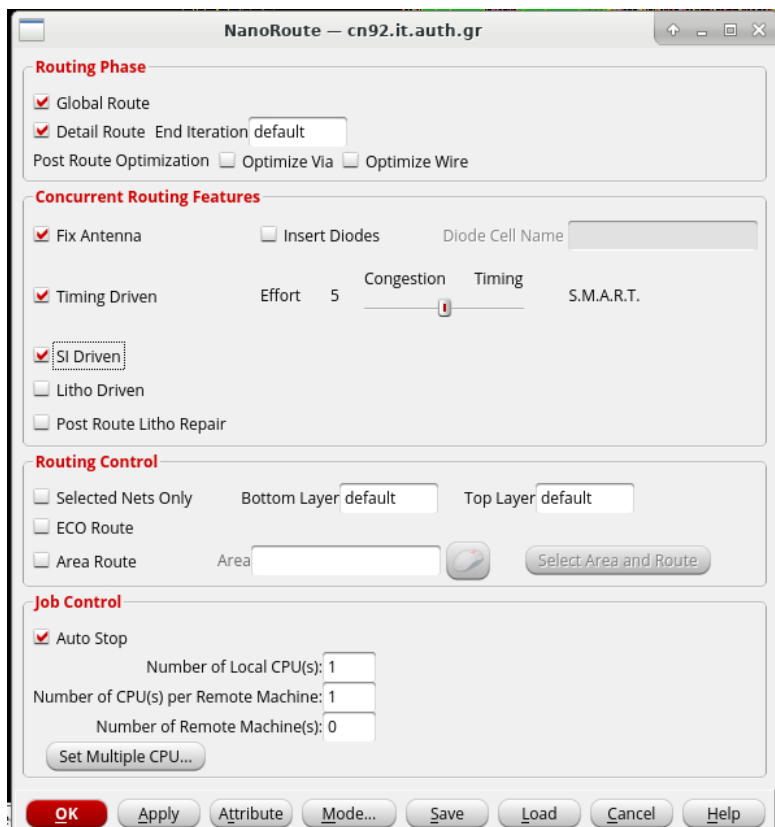


Figure 39 Exercise2 Nano Route

Μετά το πέρας της εκτέλεσης των παραπάνω επιλογών κάνουμε:

- setAnalysisMode -analysisType onChipVariation
- optDesign -postRoute

ώστε να φύγει η κάθε συμφόρηση που δημιουργείται.

Για να ελέγξουμε τα hold/setup κάνουμε τα εξής:

1. Place → Check Placement
2. Place → Refine Placement → Preserve Routing Check → OK
3. Timing → Report Timing → Post-Route & Setup or Hold

Και προκύπτει το εξής αποτέλεσμα:

```
Setup views included:
default_emulate_view
```

Setup mode	all	reg2reg	reg2cgate	default
WNS (ns):	6.316	6.501	8.224	6.316
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0
All Paths:	2342	2032	59	722

Figure 40 Exercise2 PostRoute SetupTime

Όπου μας ικανοποιούν τα αποτελέσματα για setup time το ίδιο κάνουμε και για hold time.

```
Hold views included:
default_emulate_view
```

Hold mode	all	reg2reg	reg2cgate	default
WNS (ns):	-0.032	-0.032	0.039	0.285
TNS (ns):	-0.431	-0.431	0.000	0.000
Violating Paths:	23	23	0	0
All Paths:	2342	2032	59	722

Figure 41 Exercise2 PostRoute HoldTime

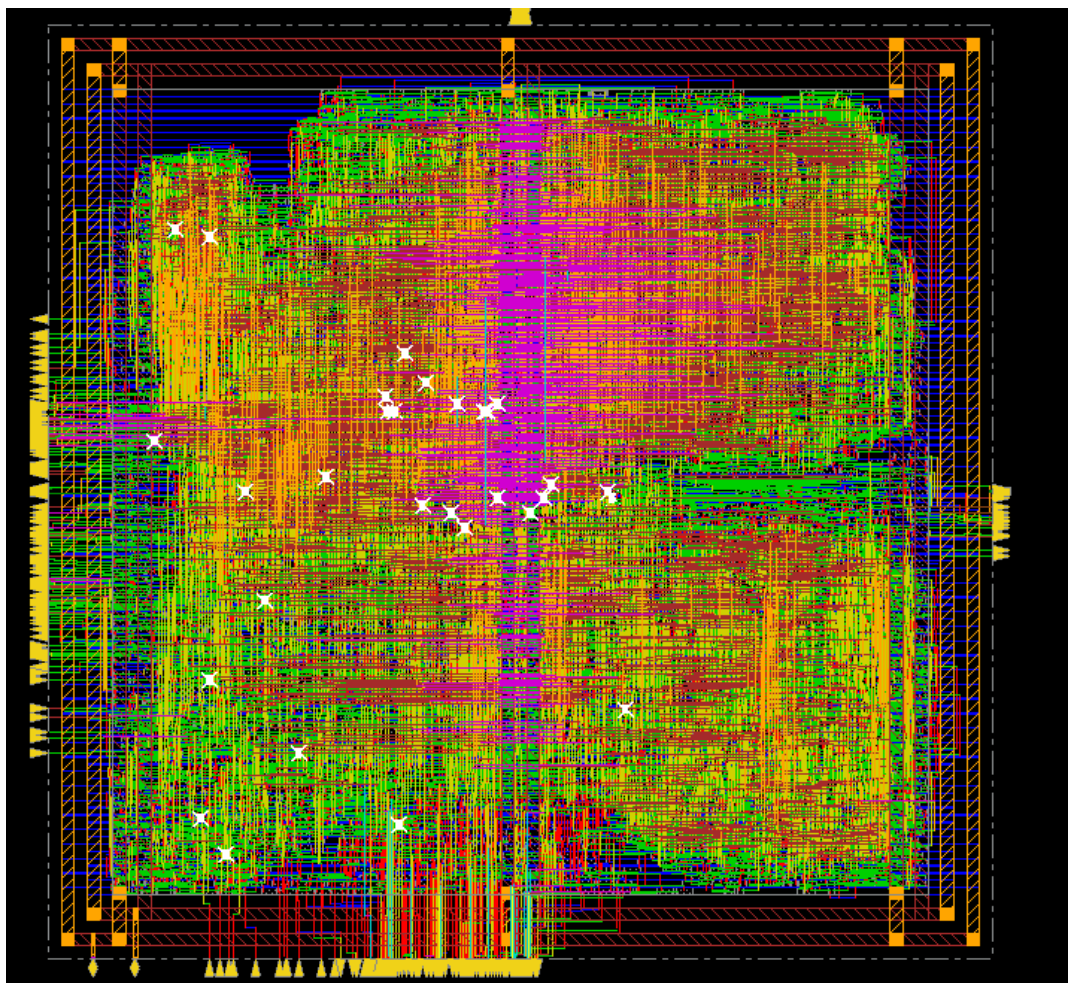
Για να φύγει κάθε περιορισμός εκτελούμε την παρακάτω εντολή στην γραμμή εντολών του Ipnovus.

- `optDesign -postRoute -setup -hold`

Και έτσι βλέπουμε την βελτίωση η οποία έγινε και είναι πολύ κοντά στο 0.

Slack	Setup time: WNS = 6.315	Hold Time: WNS= -0.002
Power	Internal Power Switching Power Leakage Power	% 24.2485 (0.679) 75.6907 (2.12) 0.0608 (0.0017)
Area	27496.116	

Figure 42 Exercise2 Physical Output



Καθώς υπάρχουν διαφορές στην ενέργεια καθώς μειώθηκε το leakage power αλλά αυξήθηκε περίπου ~2% το Switching Power καθώς και το Internal Power μειώθηκε σε ~2%. Επιπλέον αυξήθηκε και το Slack του Setup Time.

Άσκηση 3

Θα χρησιμοποιήσουμε ποσοστό χρήσης του επεξεργαστή στα 85% όπως φαίνεται παρακάτω.

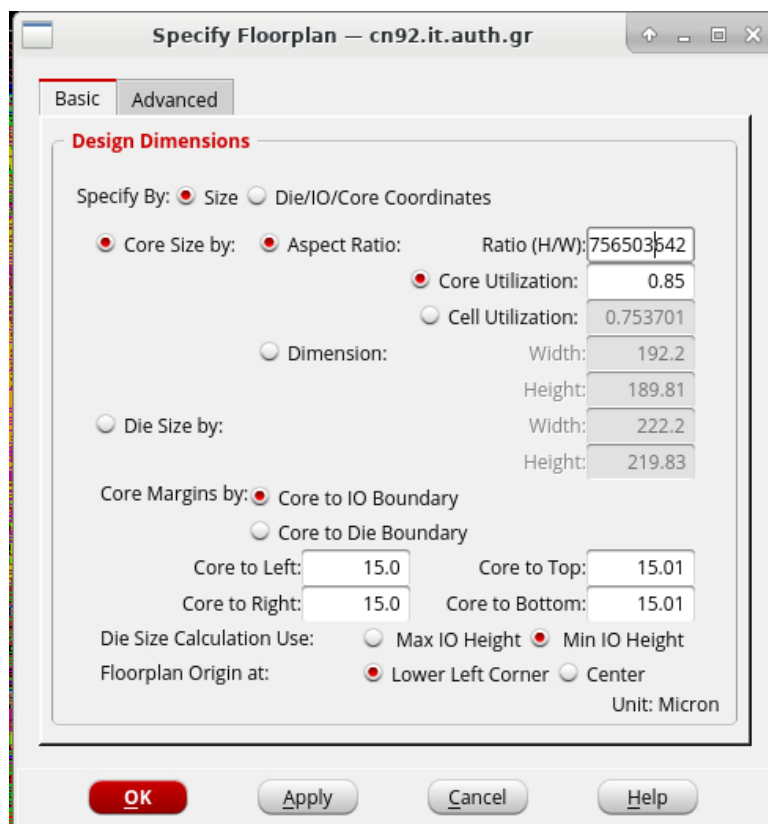


Figure 43 Exercise 3 85% Usage of CPU

Και προκύπτει:

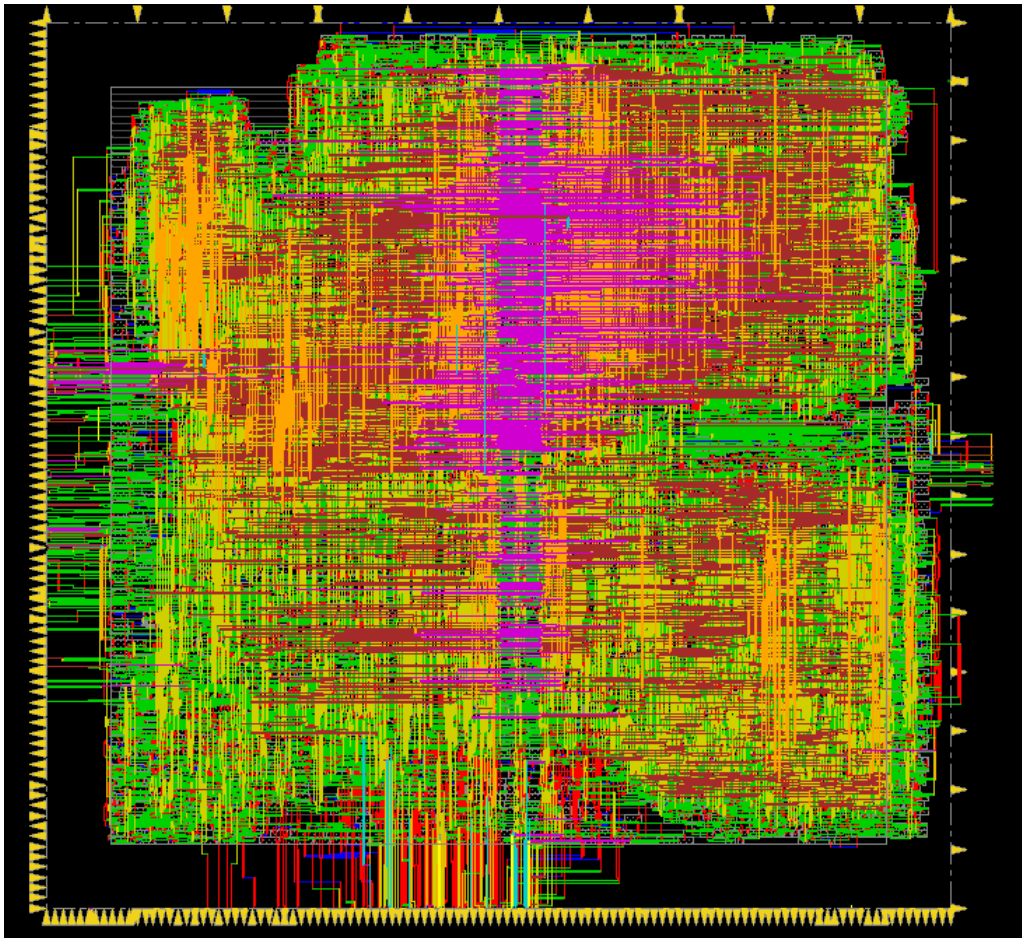


Figure 44 Exercise3 Results of 85% usage CPU

Πρώτα όμως πρέπει να ξανά δημιουργήσουμε τα Rings/Stripes.

Για το Vdd:

- `globalNetConnect Vdd -type pgpin -pin VDD -inst *`
- `globalNetConnect Vdd -type tiehi -instanceBasename *`

Για το GND:

- `globalNetConnect GND -type pgpin -pin VSS -inst *`
- `globalNetConnect GND -type tielo -instanceBasename *`

Επιπλέον πρέπει να δημιουργήσουμε(αφού τους διαγράψουμε) τους ακροδέκτες τροφοδοσίας και γείωσης στο φυσικό επίπεδο καθώς δεν υπάρχουν από πριν στο netlist με τις εξής εντολές.

- `createPGPin VDD -net Vdd -geom Metal6 20 0 20.8 0.08`
- `createPGPin GND -net GND -geom Metal7 10 0 10.8 0.08`

Και για να συνδέσουμε τα δυο επίπεδα μετάλλου (5,6) κάνουμε τα εξής:

- `editPowerVia -add_vias 1 -top_layer Metal6 -area {10 0 10.8 0.08} -bottom_layer Metal5 (για το GND)`
- `editPowerVia -add_vias 1 -top_layer Metal6 -area {20 0 20.8 0.08} -bottom_layer Metal5 (για Vdd)`

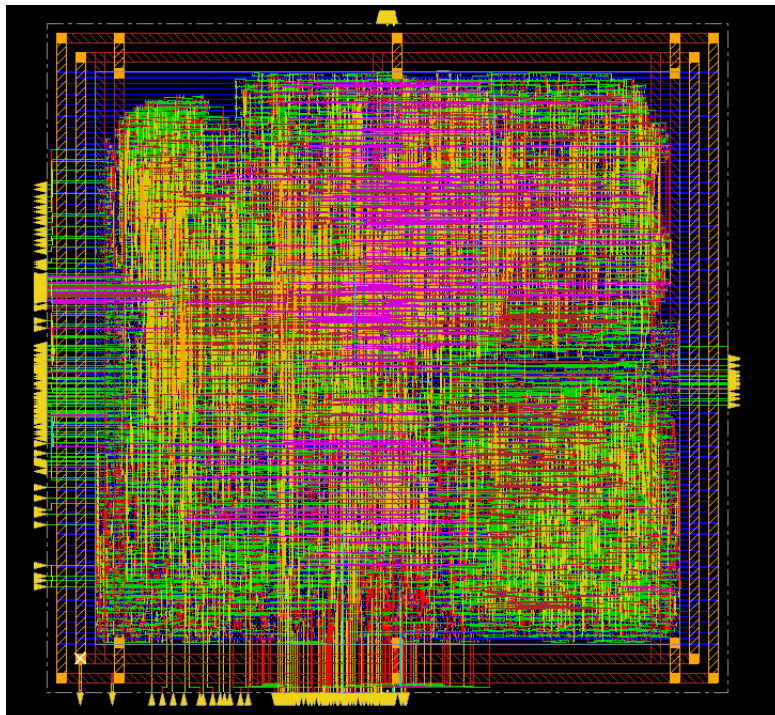
Έπειτα κάνοντας τις κατάλληλες επιλογές από το μενού Router->SRoute

Και εκτελώντας την εντολή:

- `place_opt_design`

Στη συνέχεια εκτελούμε τα εξής:

1. Place → Check Placement
2. Place → Refine Placement → Preserve Routing Check → OK
3. `optDesign`



Προκύπτουν τα εξής:

Slack	Setup time: WNS = 6.303	Hold Time: WNS = 0.007
Power	Internal Power Switching Power Leakage Power	% 24.5877 (0.6870) 75.35(2.10) 0.0619(0.0017)
Area	27915.750	

Όπου δεν υπάρχουν διαφορές.

Στη συνέχεια κάνουμε:

- Edit → Create Non Default Rules

Στη συνέχεια θα χρησιμοποιήσουμε την γραμμή εντολών με τις εξής εντολές:

- `create_route_type -top_preferred_layer 6 -bottom_preferred_layer 4 -non_default_rule exercise3 -name exercise3clock_tree`
- `set_ccopt_property target_skew 0.1`
- `set_ccopt_property target_max_trans 0.15`

Και στη συνέχεια κάνουμε:

- `delete_ccopt_clock_tree_spec`
- `create_ccopt_clock_tree_spec -file tree_step_14_exercise3.spec`

Και για να τοποθετήσουμε το δέντρο διανομής ρολογιού:

- `ccopt_design`
- `report_ccopt_clock_trees`
- `report_ccopt_skew_groups`

Number of Buffers	0
Number of Skew Group	8
Maximum depth of clock tree	1
Minimum depth of clock tree	1
Routing length for the trunk	0
Routing length for the leafes	6905.585

Όπως και πριν το εργαλείο μας δείχνει ότι τα δέντρα ρολογιού να είναι unbuffered και έτσι δεν τα αναγνωρίζει καθώς έχει μήκος των γραμμών αυτό οφείλεται στην συμπύκνωση του Chip και έτσι έχουμε μικρότερες αποστάσεις. Επιπλέον έτσι μπορεί να αυξηθεί και το Slack:

Στη συνέχεια θα εκτελέσουμε δρομολόγηση επιλέγοντας:

- Route → NanoRoute → Route
- Route → NanoRoute → Route → Mode

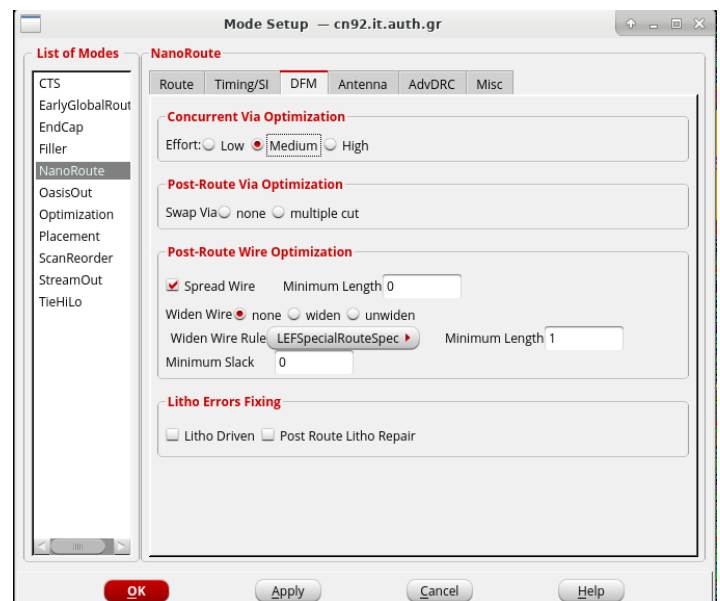
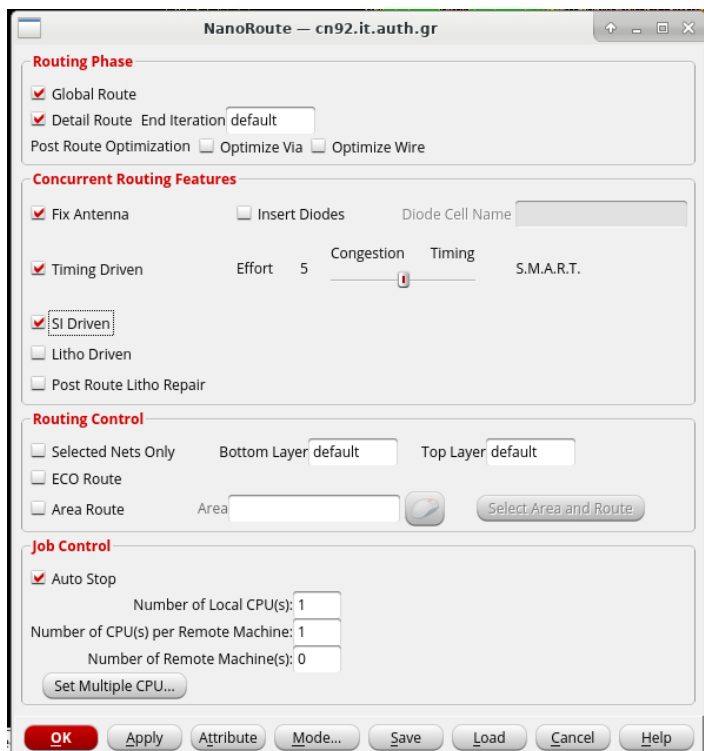


Figure 45 Exercise2 Nano Route

Μετά το πέρας της εκτέλεσης των παραπάνω επιλογών κάνουμε

- setAnalysisMode -analysisType onChipVariation
- optDesign -postRoute

ώστε να φύγει η κάθε συμφόρηση που δημιουργείται.

Για να ελέγξουμε τα hold/setup κάνουμε τα εξής:

4. Place → Check Placement
5. Place → Refine Placement → Preserve Routing Check → OK
6. Timing → Report Timing → Post-Route & Setup or Hold

Και προκύπτει το εξής αποτέλεσμα:

```
Setup views included:
default_emulate_view
```

Setup mode	all	reg2reg	reg2cgate	default
WNS (ns):	6.329	6.460	8.273	6.329
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0
All Paths:	2342	2032	59	722

Όπου μας ικανοποιούν τα αποτελέσματα για setup time το ίδιο κάνουμε και για hold time

```
Hold views included:
default_emulate_view
```

Hold mode	all	reg2reg	reg2cgate	default
WNS (ns):	-0.032	-0.032	0.031	0.285
TNS (ns):	-0.445	-0.445	0.000	0.000
Violating Paths:	25	25	0	0
All Paths:	2342	2032	59	722

Για να φύγει κάθε περιορισμός εκτελούμε την παρακάτω εντολή στην γραμμή εντολών του Innonus.

- optDesign -postRoute -setup -hold

Και έτσι βλέπουμε την βελτίωση η οποία έγινε και είναι πολύ κοντά στο 0.

Slack	Setup time: WNS = 6.327	Hold Time: WNS= -0.003
Power	Internal Power Switching Power Leakage Power	% 24.3830 (0.684) 75.555 (2.12) 0.0617 (0.0017)
Area	27970.812	

Τρέχοντας την εντολή reportRoute προκύπτει ότι:

Total length: 1.884e+05um, number of vias: 81869

M1(H) length: 1.835e+03um, number of vias: 36085

M2(V) length: 4.560e+04um, number of vias: 28521

M3(H) length: 5.643e+04um, number of vias: 10851

M4(V) length: 3.787e+04um, number of vias: 4236

M5(H) length: 2.629e+04um, number of vias: 1432

M6(V) length: 1.124e+04um, number of vias: 697

M7(H) length: 8.656e+03um, number of vias: 30

M8(V) length: 2.710e+02um, number of vias: 13

M9(H) length: 1.893e+02um, number of vias: 3

M10(V) length: 7.790e+00um, number of vias: 1

M11(H) length: 3.238e+01um

Καθώς αυξάνονται τα επίπεδα μετάλλου τα Vias μειώνονται αλλά ο μεγαλύτερος αριθμός Vias βρίσκεται στο μέταλλο 1 όπου υπάρχουν οι διασυνδέσεις μεταξύ των τρανζίστορ και είναι ευάλωτα σε συμφορήσης.

Άσκηση 4

Μετά την εκτέλεση της σύνθεσης του genus προκύπτουν τα εξής:

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
8681	39041.686	+4041	1.92337e-03

Υπάρχει μικρή μείωση των κελιών όπου αυτό οφείλεται στην μείωση του κύκλου ρολογιού από 10 ns \rightarrow 8 ns. Καθώς το εργαλείο καταβάλει προσπάθεια να επιτύχει τον απαιτούμενο χρόνο ρολογιού. Επιπλέον η επιφάνεια μειώνεται ελάχιστα αλλά και η ισχύς του κυκλώματος αυξάνεται από ~1.5 σε 1.92. Όμως την μεγάλη διαφορά κάνει το Slack από 6071 \rightarrow 4041 όπου είναι εμφανείς η δυσκολία στην αύξηση της συχνότητας του ρολογιού ενός επεξεργαστή στην σημερινή εποχή. Γι αυτό οδηγούμαστε σε διαφορετικές υλοποιήσεις παραλληλισμού υλικού για την μεγιστοποίηση της απόδοσης σε συνδυασμό με την μικρότερη δυνατή κατανάλωση ενέργειας για συγκεκριμένες εφαρμογές.

PreCTS:

Slack	Setup time: WNS = 3.522	Hold Time: WNS = 0.006
Power	Internal Power Switching Power Leakage Power	% 26.60 (0.933) 73.34(2.57) 0.0487(0.0017)
Area	26213.958	

Είναι εμφανής η διαφορά στα Slack συγκρίνοντας τον πίνακα PreCTS της ασκήσης 1. Καθώς από Setup time 6.222 \rightarrow 3.522.

PostCTS:

Slack	Setup time: WNS = 3.437	Hold Time: WNS = 0.000(μετα από <i>optDesign -postCts - hold</i>)
Power	Internal Power Switching Power Leakage Power	% 26.50 (0.93) 73.44(2.6) 0.0485(0.0017)
Area	26340.156	

Είναι εμφανής η διαφορά στα Slack συγκρίνοντας τον πίνακα PostCTS της ασκήσης 1. Καθώς από Setup time 6.179 → 3.437.

PostRoute:

Slack	Setup time: WNS = 3.421	Hold Time: WNS = - 0.003 (μετα από <i>optDesign -postRoute - hold</i>)
Power	Internal Power Switching Power Leakage Power	% 26.14 (0.93) 73.81(2.64) 0.0479(0.0017)
Area	26308.692	

Είναι εμφανής η διαφορά στα Slack συγκρίνοντας τον πίνακα PostRoute της ασκήσης 1. Καθώς από Setup time 6.158 → 3.421.

Άσκηση 5

Τρέχοντας ξανά όλα τα βήματα από το εγχειρίδιο 1 έως 16 με την βιβλιοθήκη χρονισμού `slow_vdd1v0_basicCells.lib` & σε συνδυασμό για κύκλο ρολογιού 8ns (το ίδιο αρχείο `.sdc` με την άσκηση 4) μετά την εκτέλεση της σύνθεσης του genus προκύπτουν τα εξής:

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
9754	29772.468	2	2.37448e-03

Έχουμε μια αύξηση στον αριθμό των κελιών ~1000 & αύξηση στην επιφάνεια και τεράστια διαφορά στο slack από 6071 → 2. Επιπλέον η ισχύ του κυκλώματος αυξήθηκε ~1 mW. Αυτό έγινε λόγω της αλλαγής βιβλιοθήκης από fast → slow όπου γενικότερα όλη η ανάλυση του genus έγινε πιο αργά συγκριτικά με αυτή της άσκησης 1. Καθώς η βιβλιοθήκη χρονισμού περιέχει περισσότερους παράγοντες και έτσι περισσότερη λεπτομέρεια στην σύνθεση όπου είναι πιο ακριβές μοντέλο αλλά μεγαλύτερα χρονοβόρο.

PreCTS:

Slack	Setup time: WNS = 0.429 (μετα από optDesign -preCTS)	Hold Time: WNS = 0.169
Power	Internal Power Switching Power Leakage Power	% 26.90 (0.624) 73.06(1.69) 0.0341(0.00017)
Area	30344.134	

Έχουμε μεγάλη διαφορά στα Slack για Setup time από 6.222 → 0.429 μετά από βελτιστοποίηση καθώς χωρίς αυτή θα ήταν αρνητικό. Ενώ για το Hold υπάρχει καλύτερη ανάλυση συγκριτικά με της άσκησης 1. Η ισχύς απωλειών του κυκλώματος έπεσε στο μισό συγκριτικά με την άσκηση 1.

PostCTS:

Slack	Setup time: WNS = 0.343	Hold Time: WNS = - 0.003
Power	Internal Power Switching Power Leakage Power	% 25.50 (0.92) 72.84(2.5) 0.0385(0.0017)
Area	30247.164	

Έχουμε μεγάλη διαφορά στα Slack για Setup time από 6.179 → 0.343
Εννοώ για το Hold υπάρχει καλύτερη ανάλυση συγκριτικά με της άσκησης 1. Η ισχύς απωλειών του κυκλώματος έπεσε στο μισό συγκριτικά με την άσκηση 1.

PostRoute: (μετα από *optDesign -postRoute -hold -setup*)

Slack	Setup time: WNS = 0.019	Hold Time: WNS = 0.048
Power	Internal Power Switching Power Leakage Power	% 27.15 (0.64) 72.81(1.71) 0.0340(0.0008)
Area	30392.172	

Έχουμε μεγάλη διαφορά στα Slack για Setup time από 6.158 → 0.019
μετά από βελτιστοποίηση καθώς χωρίς αυτή θα ήταν αρνητικό. Εννοώ για το Hold υπάρχει καλύτερη ανάλυση συγκριτικά με της άσκησης 1. Η ισχύς απωλειών του κυκλώματος έπεσε στο μισό συγκριτικά με την άσκηση 1.

Άσκηση 6

Τρέχοντας ξανά όλα τα βήματα από το εγχειρίδιο 1 έως 16 με clock gating(αν και εγώ σε όλες της άσκησης χρησιμοποίησα clock gating) & και με κύκλο ρολογιού 8ns (ίδιο sdc file με την άσκηση 4) μετά την εκτέλεση της σύνθεσης του genus προκύπτουν τα εξής:

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
8681	39041.686	6064	1.92337e-03

Δεν υπάρχει καμία διαφορά γιατί εξ αρχής στην υλοποίηση της άσκησης 1 χρησιμοποίησα clock gating. Παρακάτω όπου βλέπουμε ότι είχαμε αποταμίευση 82% & 87% στα RC clock instances & flip-flops

Category	Number	%	Average Toggle Saving %
RC Clock Gating Instances	58	100	81.87
RC Gated Flip-flops	1704	87	86.27
Total Ungated Flip-flops	257	13	-
Enable not found	238	93	-
Register bank width too small	19	7	-
Total Flip-flops	1961	100	-

PreCTS:

Slack	Setup time: WNS = 6.123	Hold Time: WNS = -0.034
Power	Internal Power Switching Power Leakage Power	% 26.45 (0.936) 73.49(2.60) 0.0483(0.0017)
Area	26240.292	

PostCTS:

Slack	Setup time: WNS = 6.105	Hold Time: WNS = - 0.004
Power	Internal Power Switching Power Leakage Power	% 27.10 (0.63) 72.85(1.71) 0.0339(0.00079)
Area	26350.292	

PostRoute:

Slack	Setup time: WNS = 5.877	Hold Time: WNS = - 0.032
Power	Internal Power Switching Power Leakage Power	% 25.92 (0.93) 74.02(2.67) 0.0473(0.0017)
Area	26491.044	

Άσκηση 7

Τρέχοντας ξανά όλα τα βήματα από το εγχειρίδιο 1 έως 7 όμως με την βιβλιοθήκη χρονισμού `fast_vdd1v0_basicCells.lib` το 3^ο lec από `generic_fmmap` σταματούσε το εργαλείο να λειτουργεί. Έτσι (με την βοήθεια του Αριστοτέλη υποψήφιο Διδάκτορά) καταλήξαμε ότι χρησιμοποιώντας την `slow_vdd1v0_basicCells.lib` ότι λειτουργεί ορθά για την εκτέλεση όλων των lec εντολών όπως φαίνεται παρακάτω.

RTL→elaborate:

```
lec -XL -nogui -dofile rtl_elab.do
```

...

Verification Report		

Category	Count	

1. Non-standard modeling options used:	0	
Tri-stated output:	checked	
Revised X signals set to E:	yes	
Floating signals tied to Z:	yes	
Command "add clock" for clock-gating:	not used	

2. Incomplete verification:	1	
All primary outputs are mapped:	yes	
Not-mapped DFF/DLAT is detected:	no	
All mapped points are added as compare points:	yes	
All compared points are compared:	yes	
User added black box:	no	
Black box mapped with different module name:	no	
Empty module is not black boxed:	no	
Command "add ignore outputs" used:	yes *	
Always false constraints detected:	no	
Verified pin-equivalent outputs are unmapped:	no	

3. User modification to design: 0

Change gate type: no

Change wire: no

Primary input added by user: no

4. Conformal Constraint Designer clock domain crossing checks recommended: 2

RTL5.1 Overlapped case items in parallel case statement: used *

RTL5.4 Partial case items in full case statement: used *

Multiple clocks in the design: no

5. Design ambiguity: 0

Duplicate module definition: no

Black box due to undefined cells: no

Golden design has abnormal ratio of unreachable gates: no

Ratio of golden unreachable gates: 0%

Revised design has abnormal ratio of unreachable gates: no

Ratio of revised unreachable gates: 0%

All primary input bus ordering is consistent: yes

All primary output bus ordering is consistent: yes

6. Compare Results: PASS

Total Equivalent modules = 3

pass

...

Elaborate → Generic:

`lec -XL -nogui -dofile elab_generic.do`

...

Verification Report

Category

Count

1. Non-standard modeling options used: 1

Tri-stated output:	checked
Revised X signals set to E:	yes
Floating signals tied to Z:	no *
Command "add clock" for clock-gating:	not used

2. Incomplete verification: 0

All primary outputs are mapped:	yes
Not-mapped DFF/DLAT is detected:	no
All mapped points are added as compare points:	yes
All compared points are compared:	yes
User added black box:	no
Black box mapped with different module name:	no
Empty module is not black boxed:	no
Command "add ignore outputs" used:	no
Always false constraints detected:	no

3. User modification to design: 0

Change gate type:	no
Change wire:	no
Primary input added by user:	no

4. Conformal Constraint Designer clock domain crossing checks recommended: 0

Multiple clocks in the design:	no
--------------------------------	----

5. Design ambiguity: 0

Duplicate module definition:	no
Black box due to undefined cells:	no
Golden design has abnormal ratio of unreachable gates:	no
Ratio of golden unreachable gates:	1%
Revised design has abnormal ratio of unreachable gates:	no
Ratio of revised unreachable gates:	0%
All primary input bus ordering is consistent:	yes
All primary output bus ordering is consistent:	yes

6. Compare Results: PASS

Number of EQ compare points:	2314
Number of NON-EQ compare points:	0

Number of Aborted compare points: 0

Number of Uncompared compare points : 0

=====

pass

...

Generic → Mapped

lec -XL -nogui -dofile generic_fvmap.do

...

=====

Verification Report

Category	Count
----------	-------

1. Non-standard modeling options used:	0
Tri-stated output:	checked
Revised X signals set to E:	yes
Floating signals tied to Z:	yes
Command "add clock" for clock-gating:	not used

2. Incomplete verification:	0
All primary outputs are mapped:	yes
Not-mapped DFF/DLAT is detected:	no
All mapped points are added as compare points:	yes
All compared points are compared:	yes
User added black box:	no
Black box mapped with different module name:	no
Empty module is not black boxed:	no
Command "add ignore outputs" used:	no
Always false constraints detected:	no

3. User modification to design:	0
Change gate type:	no
Change wire:	no
Primary input added by user:	no

4. Conformal Constraint Designer clock domain crossing checks recommended: 0

Multiple clocks in the design: no

5. Design ambiguity: 0

Duplicate module definition: no

Black box due to undefined cells: no

Golden design has abnormal ratio of unreachable gates: no

Ratio of golden unreachable gates: 0%

Revised design has abnormal ratio of unreachable gates: no

Ratio of revised unreachable gates: 0%

All primary input bus ordering is consistent: yes

All primary output bus ordering is consistent: yes

6. Compare Results: PASS

Number of EQ compare points: 2267

Number of NON-EQ compare points: 0

Number of Aborted compare points: 0

Number of Uncompared compare points : 0

Pass

...

Καθώς φαίνεται ότι στην αρχή αφού κάνουμε το πρώτο lsc προκύπτει ότι υπάρχουν πολλά ασύνδετα σημεία αυτό προκύπτει στην αλλαγή του αρχικού κώδικα από συμπεριφορική Verilog σε δομική Verilog ώστε να μπορέσει να αναπτυχθεί στο πυρίτιο με τις βασικές πύλες. Συγκρίνοντας τα αποτελέσματα από το rtl_elab & elab_generic βλέπουμε τις εξής διαφορές:

- Non-standard modeling options used: από 0 → 1
 - Floating signals tied to Z: από yes* → no *
- Incomplete verification: από 1 → 0
 - Command "add ignore outputs" used: από yes → no
- Conformal Constraint Designer clock domain crossing checks recommended: από 2 → 0

Δεν υπάρχει κανένα σήμα σε υψηλή σύνθετη αντίσταση καθώς δεν υπάρχουν άγνωστες έξοδοι και κανένα crossing του ρολογιού. Συγκρίνοντας τα elab_generic & generic_fmmap προκύπτουν τα εξής:

- Non-standard modeling options used: από 1 σε 0
- Design ambiguity:
 - Ratio of golden unreachable gates: από 1% σε 0%

Καθώς κάνει πιο λεπτομερή δουλειά και διώχνει και τα μικρότερα προβλήματα τις σχεδίασης.

Άσκηση 8

Χρησιμοποιώντας την βιβλιοθήκη giolib045.lef όπου περιέχει πληροφορίες για τα I/O & corner pads και το έτοιμο script pads.py όπου προκύπτουν οι δηλώσεις που πρέπει να γίνουν στο αρχείο genus.v για να μπορέσει το Innonus να τα παράγει.

Πρώτου βάλουμε τα Pads:

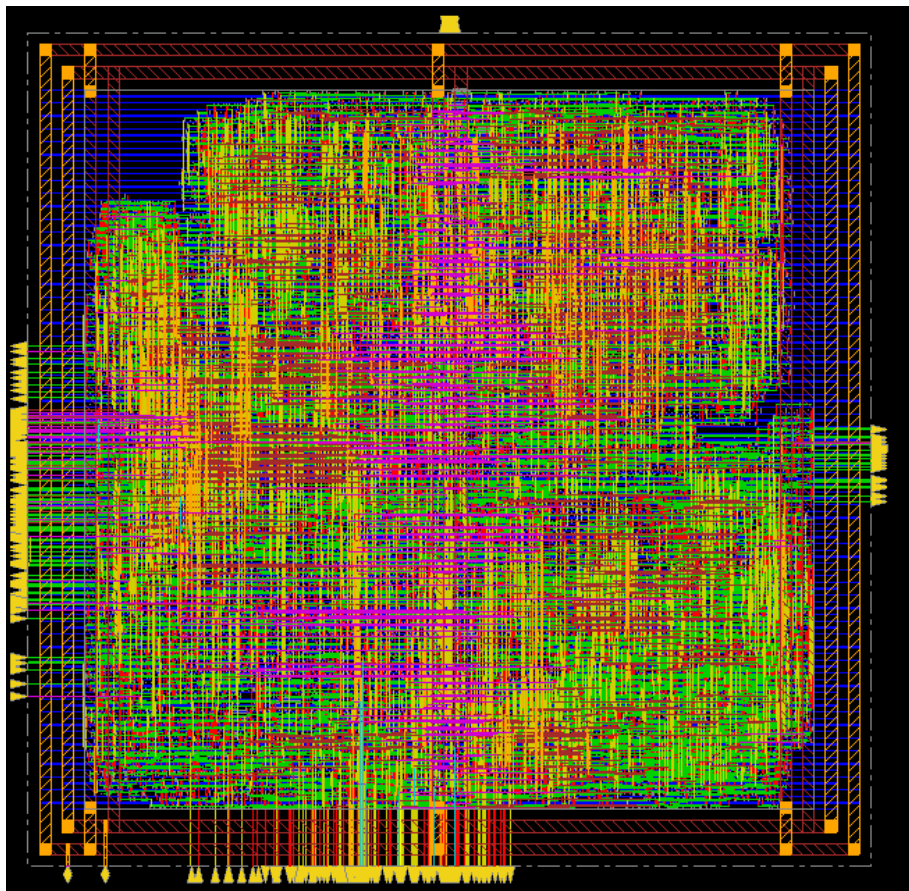


Figure 46 Exercise 8 Before Pads

Αφού βάλουμε τα Pads:

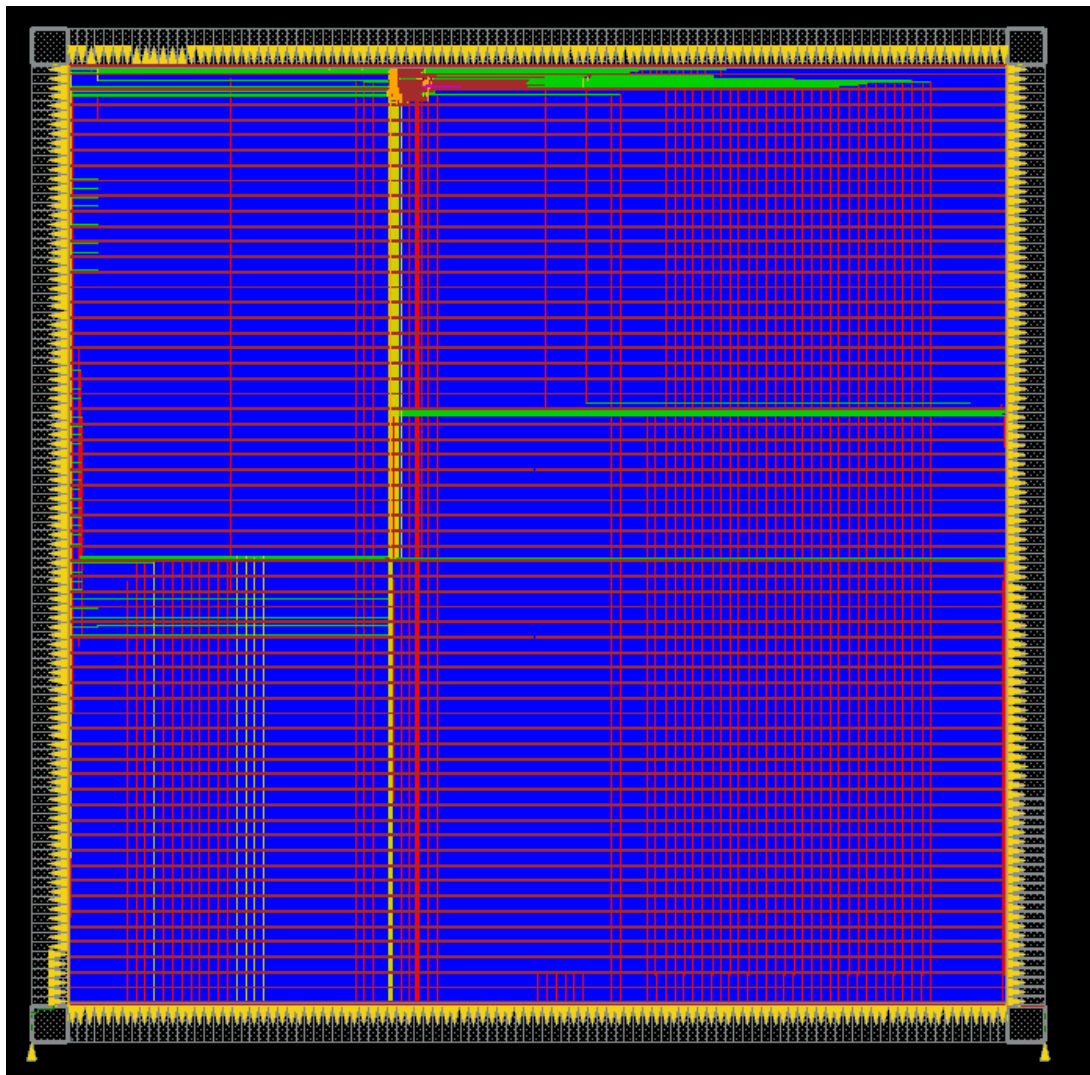


Figure 47 Exercise 8 After Pads

Total Area with Pads + CPU	6147380.178 μm
Only CPU	27380.178 μm

Όπως φαίνεται από τον παραπάνω πίνακα η επιφάνεια που καταλαμβάνουν τα Pads αποτελείται από 6120000 μm επιφάνεια. Επιπλέον υπάρχει διαφορά στον τρόπο με τον οποίο δρομολογεί όταν υπάρχουν τα pads και όχι μόνο τα Pins καθώς περιέχει περισσότερη πληροφορία στην διασύνδεση έτσι δεν μπορεί να προσφέρει την ίδια λεπτομέρεια. Πρόσθετα τα Pins τροφοδοσίας (VDD/GND ή VDD/VSS) τοποθετήθηκαν στα corner pads στα

κάτω αριστερά(VSS) και δεξιά (VDD). Γενικότερα όλη η εκτέλεση του προγράμματος κράτησε πολύ περισσότερη ώρα συγκριτικά με την άσκηση 1.

Άσκηση 9

Με την εντολή **check_dft_rules** βλέπουμε τον έλεγχο της εισαγωγής της λογικής DFT όπου προκύπτει ότι όλοι οι registers είναι ανιχνεύσιμη

- Number of usable scan cells: 48
- Total number of Test Clock Domains: 1
DFT Test Clock Domain: scanclk
Test Clock 'scanclk' (Positive edge) has 2090 registers
Number of user specified non-Scan registers: 0
Number of registers that fail DFT rules: 0
Number of registers that pass DFT rules: 2090
Percentage of total registers that are scannable: 100%

Στη συνέχεια βλέπουμε την εντολή **report_scan_registers**.

Summary:

Total registers that pass DFT rules: 2090

Total registers that fail DFT rules: 0

Total registers that are marked preserved or dont-scan: 0

Total registers that are marked Abstract Segment dont-scan: 0

Total registers that are part of shift register segments: 0

Total registers that are lockup elements: 0

Total registers that are level-sensitive: 0

Total registers that are misc. non-scan: 0

Βλέπουμε ότι και οι 2090 registers έκαναν PASS στους κανόνες που ορίσαμε.

Έπειτα βλέπουμε την εντολή **report_scan_setup** όπου δεν προκύπτει κάποιο πρόβλημα.

Μετά το syn_gen.

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
22429	73335.558	6917	1.07238e-03

Μεγάλη αύξηση της επιφάνειας λόγο της έξτρα λογικής που προσθέτουμε με το DFT rules.

Μετά το syn_generic.

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
22322	73003.614	6904	1.07749e-03

Μεγάλη αύξηση της επιφάνειας λόγο της έξτρα λογικής που προσθέτουμε με το DFT rules.

Μετά το syn_map.

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
8902	39212.943	6118	2.20562e-04

Μετά το syn_opt.

Αριθμός Κελίων	Επιφάνεια	Slack	Ισχύ κυκλώματος
8689	39012.960	6071	2.20407e-04

Όπως αρχικά ήταν και χωρίς έλεγχο DFT.

Μετά την εκτέλεση της εντολής **check_dft_rules -advanced** προκύπτει ότι:

Total number of DFT violations: 0

Total number of Test Clock Domains: 1

DFT Test Clock Domain: scanclk

Test Clock 'scanclk' (Positive edge) has 2090 registers

Number of user specified non-Scan registers: 0

Number of registers that fail DFT rules: 0

Number of registers that pass DFT rules: 2090

Percentage of total registers that are scannable: 100%

Όπως και πριν δεν υπάρχει κανένα fail στα DFT rules.