

# POCOP マクロ、Astro、ClubLayout(ふんが)、2010.5.10

## 0.1 はじめに

POCOP\_TOP は、パイプライン化した POCO のコアのみをレイアウトしたものである。階層化されていて、まず、実体である POCOP をレイアウトし、これを POCOP\_TOP のど真ん中に置いて全体をレイアウトする作戦を取っている。本ドキュメントは、このマクロ部分に関するものである。hunga/verilog/f65/pr にレイアウト用のディレクトリがある。合成時に使ったディレクトリは、hunga/verilog/f65/syn であるが、合成についての解説は含まれていない。

## 0.2 ディレクトリセットアップ

以下のファイルはどこに置いておいても良いがやはり適当に整理して置くのが良いと思う。lib などは f65 で共通なので、どこかのをリンクしても良い。

- CS202IO(I/O ライブラリ、リンクする。今回は使わないが POCOP\_TOP 部を作る際に必要) : /home/vdec/lib/fujitsu65/milky/data/CS202IO/lib/CS202IO
- CS202SZ(Cell ライブラリ、リンクする) : /home/vdec/lib/fujitsu65/milky/data/CS202IO/lib/CS202SZ/
- lib: f65.tf, gdsout.map, tlu2mw.map, tlu\_plus\_mfe.best, tlu\_plus\_mfe.worst を入れておく
- tdf: POCOP.tdf を入れておく
- vnet: 合成後の verilog 記述、POCOP.vnet, POCOP\_TOP.vnet を入れておく
- sdc: 合成時に生成した POCOP.sdc を入れておく
- scripts: tcl ファイルを入れておく。
- postlayout\_net: レイアウト後の verilog ファイルを入れる所。
- rpt: レポートファイルを入れる所。

注意 : Astro はファイルを吐き出すディレクトリが存在しないと、何も言わずに落ちる場合がある。

## 1 POCOP.tcl の解説

Astro を立ち上げて、tcl の所をクリックし、source POCOP.tcl とやると以下が実行され、POCOP のレイアウトが生成される。これを順に解説する。

### 1.1 初期設定

以下、利用する変数を定義する。コアのサイズはいい加減に決めた。100um X 100um で十分収まる。

```
set design_name POCOP
set is_combinational_circuit false

set core_width 100
set core_height 100
set cell_width 101.8
```

```

set cell_height 101.4

set core_to_top    0.9
set core_to_bottom 0.9
set core_to_right  0.9
set core_to_left   0.9

set ref_libs {"CS202SZ"}

source ./scripts/verilog_to_cell.tcl

```

## 1.2 verilog file の読み込み

読み込み中 0 / 1 の変換、テクノロジファイルの指定等を行う。

```

auVerilogToCell
setFormField verilog_to_cell library_name           ${design_name}
setFormField verilog_to_cell verilog_file_name     ../../syn/vnet/${design_name}.vnet
setFormField verilog_to_cell output_cell_name      ${design_name}
setFormField verilog_to_cell top_module_name       ${design_name}
setFormField verilog_to_cell tech_file_name        ./lib/f65.tf
setFormField verilog_to_cell net_name_for_1'b0     VSS
setFormField verilog_to_cell net_name_for_1'b1     VDD
setFormField verilog_to_cell set_case_sensitive    1
setFormField verilog_to_cell open_library_and_cell_when_done 1

formButton verilog_to_cell refLibOptions
foreach ref_lib $ref_libs {
setFormField verilog_to_cell reference_library     $ref_lib
formButton verilog_to_cell add
}
subFormHide verilog_to_cell 2

formButton "Verilog To Cell" "globalNetOptions"
setFormField "Verilog To Cell" "Net Name" "VDD"
setFormField "Verilog To Cell" "Port Pattern" "VDD"
formButton "Verilog To Cell" "apply"
setFormField "Verilog To Cell" "Net Name" "VSS"
setFormField "Verilog To Cell" "Port Pattern" "VSS"
formButton "Verilog To Cell" "apply"
subFormHide "Verilog To Cell" 1
formOK verilog_to_cell

```

レイアウトウィンドウが生成される。

```

source ./scripts/apply_tlu_plus.tcl

```

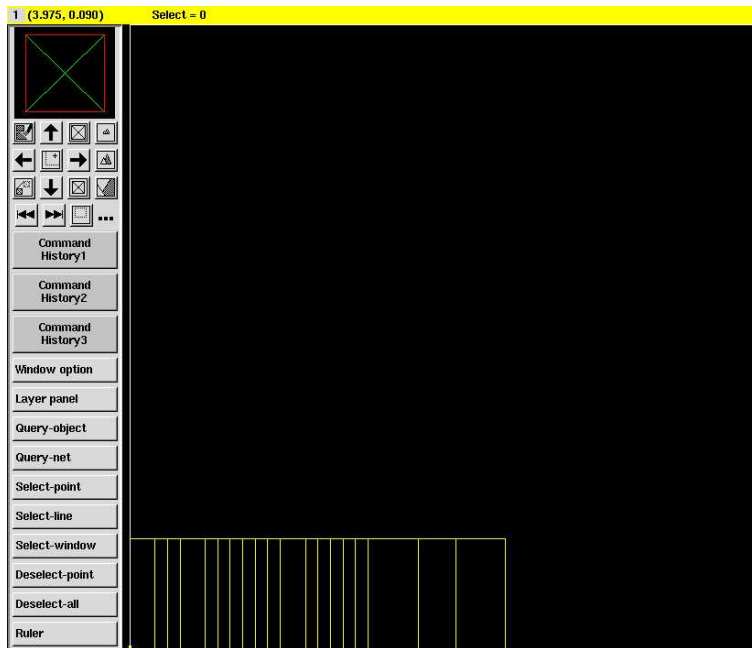


図 1: 読み込んだ後

各ライブラリの標準セルの ITF を TLU+という形式に変換し、タイミングドリブン設計ができるようにする。要するに遅延のテーブルみたいなもの。

```
cmItfToTLUPlus
```

```
setFormField "Conver ITF to TLU+" "Library Name" ${design_name}
setFormField "Convert ITF to TLU" "MIN" "1"
setFormField "Convert ITF to TLU" "NOM" "0"
setFormField "Convert ITF to TLU" "MAX" "1"
setFormField "Convert ITF to TLU" "Min CapTable File" "./lib/tlu_plus_mfe.best"
setFormField "Convert ITF to TLU" "Max CapTable File" "./lib/tlu_plus_mfe.worst"
setFormField "Convert ITF to TLU" "Star-RCXT Mapping File" "./lib/tlu2mw.map"
formOK "Convert ITF to TLU"
```

```
atTimingSetup
```

```
atTimingSetupGoto "Parasitics"
atCmdSetField "Parasitic Model Operating Conditions" "max min"
atCmdSetField "Parasitic Model Capacitance Model" "tluplus"
atCmdSetParaModel
atTimingSetupHide
```

### 1.3 フロアプラン

まず、tdfを読み込む

```
source ./scripts/load_tdf.tcl
```

tdfとはterminal definition file といってピン配置を指定するファイル。これは以下の形式で適当に配置した。ここで、座標はすべて0.1としたが、これで適当に配置してくれる。

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; PE
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

;; TOTAL :
;; NORTH : 35
;; SOUTH : 111

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; NORTH (35) ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

pin "IDATAIN[0]"      "MET4"  0.1  0.1  "TOP"    1
pin "IDATAIN[1]"      "MET4"  0.1  0.1  "TOP"    2
pin "IDATAIN[2]"      "MET4"  0.1  0.1  "TOP"    3
pin "IDATAIN[3]"      "MET4"  0.1  0.1  "TOP"    4
pin "IDATAIN[4]"      "MET4"  0.1  0.1  "TOP"    5
pin "IDATAIN[5]"      "MET4"  0.1  0.1  "TOP"    6
pin "IDATAIN[6]"      "MET4"  0.1  0.1  "TOP"    7
pin "IDATAIN[7]"      "MET4"  0.1  0.1  "TOP"    8
pin "IDATAIN[8]"      "MET4"  0.1  0.1  "TOP"    9
pin "IDATAIN[9]"      "MET4"  0.1  0.1  "TOP"   10
pin "IDATAIN[10]"     "MET4"  0.1  0.1  "TOP"   11
pin "IDATAIN[11]"     "MET4"  0.1  0.1  "TOP"   12
pin "IDATAIN[12]"     "MET4"  0.1  0.1  "TOP"   13
pin "IDATAIN[13]"     "MET4"  0.1  0.1  "TOP"   14
pin "IDATAIN[14]"     "MET4"  0.1  0.1  "TOP"   15
pin "IDATAIN[15]"     "MET4"  0.1  0.1  "TOP"   16

```

```

pin "IADDR[0]"        "MET4"  0.1  0.1  "TOP"   17
pin "IADDR[1]"        "MET4"  0.1  0.1  "TOP"   18
pin "IADDR[2]"        "MET4"  0.1  0.1  "TOP"   19
pin "IADDR[3]"        "MET4"  0.1  0.1  "TOP"   20
pin "IADDR[4]"        "MET4"  0.1  0.1  "TOP"   21
pin "IADDR[5]"        "MET4"  0.1  0.1  "TOP"   22
pin "IADDR[6]"        "MET4"  0.1  0.1  "TOP"   23
pin "IADDR[7]"        "MET4"  0.1  0.1  "TOP"   24
pin "IADDR[8]"        "MET4"  0.1  0.1  "TOP"   25
pin "IADDR[9]"        "MET4"  0.1  0.1  "TOP"   26
pin "IADDR[10]"       "MET4"  0.1  0.1  "TOP"   27
pin "IADDR[11]"       "MET4"  0.1  0.1  "TOP"   28
pin "IADDR[12]"       "MET4"  0.1  0.1  "TOP"   29
pin "IADDR[13]"       "MET4"  0.1  0.1  "TOP"   30
pin "IADDR[14]"       "MET4"  0.1  0.1  "TOP"   31
pin "IADDR[15]"       "MET4"  0.1  0.1  "TOP"   32

```

```

pin "CLK"          "MET4" 0.1 0.1 "TOP"    33
pin "RST_N"        "MET4" 0.1 0.1 "TOP"    34
pin "WE_N"         "MET4" 0.1 0.1 "TOP"    35

```

```

;;;;;;; SOUTH (111) ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

pin "DDATAIN[0]"   "MET4" 0.1 0.1 "BOTTOM" 1
pin "DDATAIN[1]"   "MET4" 0.1 0.1 "BOTTOM" 2
pin "DDATAIN[2]"   "MET4" 0.1 0.1 "BOTTOM" 3
pin "DDATAIN[3]"   "MET4" 0.1 0.1 "BOTTOM" 4
pin "DDATAIN[4]"   "MET4" 0.1 0.1 "BOTTOM" 5
pin "DDATAIN[5]"   "MET4" 0.1 0.1 "BOTTOM" 6
pin "DDATAIN[6]"   "MET4" 0.1 0.1 "BOTTOM" 7
pin "DDATAIN[7]"   "MET4" 0.1 0.1 "BOTTOM" 8
pin "DDATAIN[8]"   "MET4" 0.1 0.1 "BOTTOM" 9
pin "DDATAIN[9]"   "MET4" 0.1 0.1 "BOTTOM" 10
pin "DDATAIN[10]"  "MET4" 0.1 0.1 "BOTTOM" 11
pin "DDATAIN[11]"  "MET4" 0.1 0.1 "BOTTOM" 12
pin "DDATAIN[12]"  "MET4" 0.1 0.1 "BOTTOM" 13
pin "DDATAIN[13]"  "MET4" 0.1 0.1 "BOTTOM" 14
pin "DDATAIN[14]"  "MET4" 0.1 0.1 "BOTTOM" 15
pin "DDATAIN[15]"  "MET4" 0.1 0.1 "BOTTOM" 16

```

```

pin "DDATAOUT[0]"  "MET4" 0.1 0.1 "BOTTOM" 17
pin "DDATAOUT[1]"  "MET4" 0.1 0.1 "BOTTOM" 18
pin "DDATAOUT[2]"  "MET4" 0.1 0.1 "BOTTOM" 19
pin "DDATAOUT[3]"  "MET4" 0.1 0.1 "BOTTOM" 20
pin "DDATAOUT[4]"  "MET4" 0.1 0.1 "BOTTOM" 21
pin "DDATAOUT[5]"  "MET4" 0.1 0.1 "BOTTOM" 22
pin "DDATAOUT[6]"  "MET4" 0.1 0.1 "BOTTOM" 23
pin "DDATAOUT[7]"  "MET4" 0.1 0.1 "BOTTOM" 24
pin "DDATAOUT[8]"  "MET4" 0.1 0.1 "BOTTOM" 25
pin "DDATAOUT[9]"  "MET4" 0.1 0.1 "BOTTOM" 26
pin "DDATAOUT[10]" "MET4" 0.1 0.1 "BOTTOM" 27
pin "DDATAOUT[11]" "MET4" 0.1 0.1 "BOTTOM" 28
pin "DDATAOUT[12]" "MET4" 0.1 0.1 "BOTTOM" 29
pin "DDATAOUT[13]" "MET4" 0.1 0.1 "BOTTOM" 30
pin "DDATAOUT[14]" "MET4" 0.1 0.1 "BOTTOM" 31
pin "DDATAOUT[15]" "MET4" 0.1 0.1 "BOTTOM" 32

```

```

pin "DADDR[0]"     "MET4" 0.1 0.1 "BOTTOM" 33
pin "DADDR[1]"     "MET4" 0.1 0.1 "BOTTOM" 34
pin "DADDR[2]"     "MET4" 0.1 0.1 "BOTTOM" 35
pin "DADDR[3]"     "MET4" 0.1 0.1 "BOTTOM" 36
pin "DADDR[4]"     "MET4" 0.1 0.1 "BOTTOM" 37
pin "DADDR[5]"     "MET4" 0.1 0.1 "BOTTOM" 38

```

```

pin "DADDR[6]"           "MET4" 0.1 0.1 "BOTTOM" 39
pin "DADDR[7]"           "MET4" 0.1 0.1 "BOTTOM" 40
pin "DADDR[8]"           "MET4" 0.1 0.1 "BOTTOM" 41
pin "DADDR[9]"           "MET4" 0.1 0.1 "BOTTOM" 42
pin "DADDR[10]"          "MET4" 0.1 0.1 "BOTTOM" 43
pin "DADDR[11]"          "MET4" 0.1 0.1 "BOTTOM" 44
pin "DADDR[12]"          "MET4" 0.1 0.1 "BOTTOM" 45
pin "DADDR[13]"          "MET4" 0.1 0.1 "BOTTOM" 46
pin "DADDR[14]"          "MET4" 0.1 0.1 "BOTTOM" 47
pin "DADDR[15]"          "MET4" 0.1 0.1 "BOTTOM" 48

```

以下は社長のアンテナ防止策でなくてもいいとのこと。

```

source ./scripts/pin_guide.tcl
source ./scripts/antenna_route_guide.tcl

```

これがフロアプラン、コアの大きさ、周辺との幅を定める。

```

source ./scripts/floorplan.tcl

```

中身は以下の通り

```

axgPlanner

```

```

setFormField floor_planning control_parameter      "width & height"
setFormField floor_planning row/core_ratio         1.0
setFormField floor_planning core_width             ${core_width}
setFormField floor_planning core_height            ${core_height}

setFormField floor_planning horizontal_row         1
setFormField floor_planning double_back            1
setFormField floor_planning start_from_first_row   0
setFormField floor_planning flip_first_row         0

setFormField floor_planning core_to_top            $core_to_top
setFormField floor_planning core_to_bottom         $core_to_bottom
setFormField floor_planning core_to_left           $core_to_left
setFormField floor_planning core_to_right          $core_to_right

setFormField "Floor Planning" "Max Metal Routing Layer" "11"

```

```

formOK floor_planning

```

これで、POCOP のコアが表示される。

以下は、後始末と、チェック。これで第一階は終わりひとまず結果を書き込む。

```

source ./scripts/check_design.tcl
source ./scripts/remove_blockage.tcl
source ./scripts/remove_route_guide.tcl
source ./scripts/check_timing.tcl

```

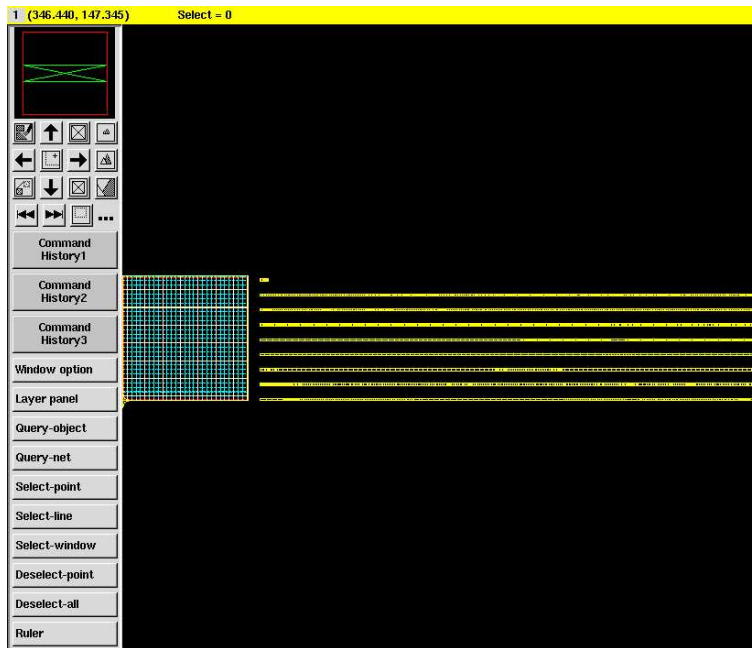


図 2: フロアプラン後

以下は、電源、グラウンドの接続で節目毎にやっておく必要がある。

```
source ./scripts/connect_pg.tcl
source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "01_pre_route_pg"
formOK "Save As"
```

## 1.4 ストラップとレール

第2段階はストラップとレール、ここではストラップは縦方向だけにし、MET6を使う。間隔は社長が使ったままにしてある。

```
source ./scripts/route_strap.tcl
-----中身-----
axgDelRouteType
formDefault delete_route_type
setToggleField delete_route_type p/g strap 1
formOK delete_route_type

axgCreateStraps
setFormField "Create Straps" "Direction" "Vertical"
setFormField "Create Straps" "Start X" "3.6"
setFormField "Create Straps" "Net Name(s)" "VDD"
setFormField "Create Straps" "Layer" "MET6"
setFormField "Create Straps" "Width" "1.8"
setFormField "Create Straps" "Configure by" "Step & Stop"
setFormField "Create Straps" "Step" "43.2"
```

```

setFormField "Create Straps" "Stop" $cell_width
setFormField "Create Straps" "Pitch within Group" "43.2"
setFormField "Create Straps" "Low Ends" "At Core Bdry"
setFormField "Create Straps" "Extend to Low Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Force Lo" "1"
setFormField "Create Straps" "High Ends" "At Core Bdry"
setFormField "Create Straps" "Extend to High Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Force Hi" "1"
formApply "Create Straps"

setFormField "Create Straps" "Direction" "Vertical"
setFormField "Create Straps" "Start X" "25.2"
setFormField "Create Straps" "Net Name(s)" "VSS"
setFormField "Create Straps" "Layer" "MET6"
setFormField "Create Straps" "Width" "1.8"
setFormField "Create Straps" "Configure by" "Step & Stop"
setFormField "Create Straps" "Step" "43.2"
setFormField "Create Straps" "Stop" $cell_width
setFormField "Create Straps" "Pitch within Group" "43.2"
setFormField "Create Straps" "Low Ends" "At"
setFormField "Create Straps" "Extend to Low Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Force Lo" "1"
setFormField "Create Straps" "High Ends" "At Core Bdry"
setFormField "Create Straps" "Extend to High Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Force Hi" "1"
formOK "Create Straps"

```

ルールは横方向のみ、セルの電源、グラウンドの MET1 が普通に使われるはず。

```

source ./scripts/route_rail.tcl
-----中身-----
axgPrerouteStandardCells
formDefault preroute_standard_cells
setFormField preroute_standard_cells "Fill All Empty Rows" "1"
setFormField preroute_standard_cells "extend_to_boundaries_and_generate_pins" "1"
setToggleField preroute_standard_cells "Do Not Connect" "Macro Pins" "0"
setFormField preroute_standard_cells do_not_route_over_macros 0
setFormField "Preroute Standard Cells" "Extend to Boundaries and Generate Pins" "1"
setFormField "Preroute Standard Cells" "Force" "1"

formButton preroute_standard_cells DRC
setFormField preroute_standard_cells protect_signal_pin_access_edges 1
subFormHide preroute_standard_cells 1

setFormField preroute_standard_cells select_pins_automatically_and_route All
formOK preroute_standard_cells

```

ここでも、電源とグラウンドを接続しておく。



```
source ./scripts/connect_pg.tcl
```

ここまでで、縦横の電源グリッドができているのがわかる。

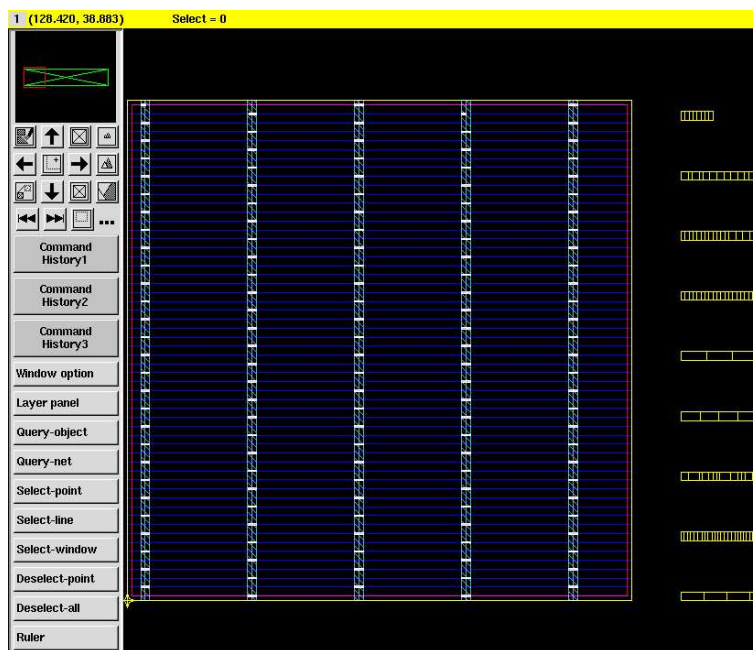


図 3: ストラップ/レール

ここで、タップを行単位で一定間隔で入れてやる。昔のセルは、それぞれのセル中で電源、グランドに繋いでいたが、最近のセルは面積の節約のために、一定間隔でタップセルを入れて接続する。このため、これがないと、セルは全く動作しないので、注意。DRC.ERC の原因になる。これは一定間隔で絶対入れないとまずいので、プレースの前に入れてやる。

```
source ./scripts/insert_tap_array.tcl
```

-----中身-----

```
axSetIntParam "apl" "tapCellUniformPlacement" 1
```

```
axgArrayTapCell
```

```
formDefault array_tap_cell
```

```
setFormField array_tap_cell tap_master_name "SC23YUZTAP021"
```

```
setFormField array_tap_cell pattern Normal
```

```
setFormField array_tap_cell tap_cell_distance_in_array 36
```

```
setToggleField array_tap_cell no_tap_under_mx m1 1
```

```
setToggleField array_tap_cell no_tap_under_mx m2 1
```

```
setFormField array_tap_cell connect_to_power_net_(optional) VDD
```

```
setFormField array_tap_cell connect_to_ground_net_(optional) VSS
```

```
setFormField "Array Tap Cell" "Name Separator (optional)" "_"
```

```
formOK array_tap_cell
```

```
axAddEndCap [geGetEditCell] SC23YUZTAP021 0 0 1 0
```

入れる間隔は DRC/ERC エラーが起きない推奨値を使っている。以下の黄色く縦に入った箱がタップ  
以下は、社長のアンテナ止め。

```
source ./scripts/antenna_route_guide.tcl
```

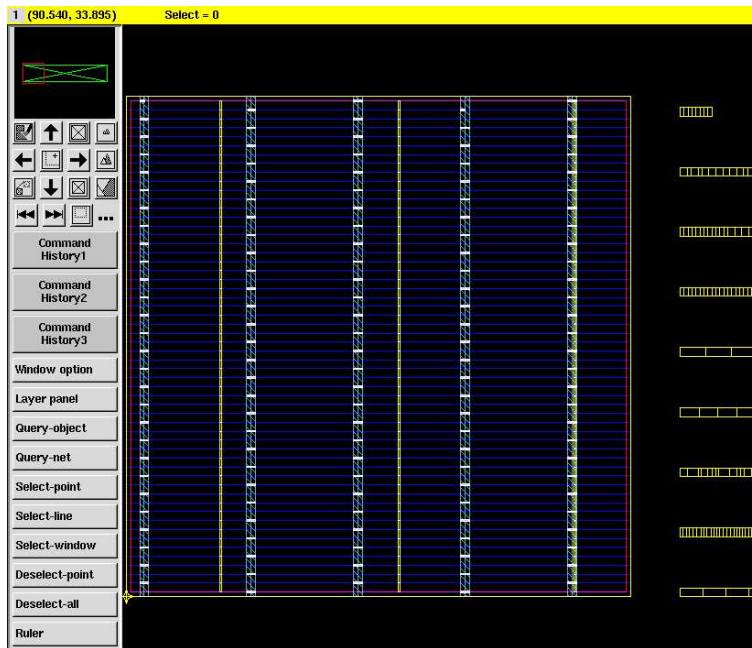


図 4: タップのアレイ

```
source ./scripts/connect_pg.tcl
```

```
source ./scripts/save_cell.tcl
```

```
geSaveAs
```

```
setFormField "Save As" "Cell Name" "02_pre_auto_place"
```

```
formOK "Save As"
```

## 1.5 プレース

これよりプレースを行う。まずは sdc をロードする。

```
source ./scripts/load_sdc.tcl
```

```
-----中身-----
```

```
ataRemoveTC
```

```
ataLoadSDC
```

```
formDefault load_sdc_file
```

```
setFormField load_sdc_file sdc_file_name ../../syn/sdc/${design_name}.sdc
```

```
formOK load_sdc_file
```

```
ataWriteTC
```

```
setFormField write_timing_constraint file_name ./sdc/${design_name}.sdc
```

```
formOK write_timing_constraint
```

クロックツリーのセットアップを行う。

```
source ./scripts/pre_cts_timing_setup.tcl
```

```
-----中身-----
```

```
atTimingSetup
```

```
atTimingSetupGoto "Model1"
```

```

atCmdSetField delay_model_net_delay_model          medium_effort
atCmdSetField delay_model_cell_delay_opcond        "max min"
atCmdSetModels

```

```

atTimingSetupGoto                                  "Environment"
atCmdSetField ignore_interconnect                  1
atCmdSetField ignore_propagated_clock              1
atCmdSetField enable_mixed_clock/signal_edges      0
atCmdSetField enable_gated_clock_checks            0
atCmdSetEnvModel

```

```

atTimingSetupGoto                                  "Optimization"
atCmdSetField optimization_max_capacitance          0
atCmdSetField optimization_max_transition           0
atCmdSetField optimization_target_setup_slack       0.1
atCmdSetOptModel

```

```

atTimingSetupGoto                                  "Parasitics"
atCmdSetField parasitic_model_operating_conditions "max min"
atCmdSetParaModel

```

```

atTimingSetupHide

```

ここで、いよいよプレースメントをする。

```

source ./scripts/auto_place.tcl
-----中身-----
axSetIntParam "pds" "more_loc_in_ppo" 1

```

```

pdsHFNCollapse 10

```

```

astPlaceOptions
formDefault astroplace_options
setFormField astroplace_options cong_driven          1
setFormField astroplace_options timing_driven         1
setFormField astroplace_options plan_group            1
setToggleField astroplace_options short_checking_at_mx m1 1
setToggleField astroplace_options short_checking_at_mx m2 1
setToggleField astroplace_options short_checking_at_mx m3 1
setToggleField astroplace_options short_checking_at_mx m4 1
setToggleField astroplace_options short_checking_at_mx m5 1
setToggleField astroplace_options short_checking_at_mx m6 1
setToggleField astroplace_options short_checking_at_mx m7 1
setToggleField astroplace_options short_checking_at_mx m8 1
setToggleField astroplace_options short_checking_at_mx m9 1
setToggleField astroplace_options short_checking_at_mx m10 1
setToggleField astroplace_options short_checking_at_mx m11 1

```

```
formOK astroplace_options
```

```
astAutoPlace
```

```
formDefault auto_place
```

```
#setFormField auto_place post_place_optimization 0
```

```
setFormField auto_place effort medium
```

```
setFormField auto_place place_prevent_xtalk 1
```

```
formButton auto_place detailOption
```

```
setFormField auto_place opt_prevent_xtalk 1
```

```
formOK auto_place
```

```
astSearchRefine
```

```
formDefault place_cells_-_search_&_refine
```

```
setFormField place_cells_-_search_&_refine speed_per_area: medium
```

```
formOK place_cells_-_search_&_refine
```

ここで、再び電源を接続

```
\begin{verbatim}
```

```
source ./scripts/connect_pg.tcl
```

これで、横っちょにあった黄色い箱が中に配置されるはず。

以下は再び社長のアンテナ止め、その後にピンの最適化をやって保存

```
source ./scripts/pin_guide.tcl
```

```
source ./scripts/pin_loc_opt.tcl
```

```
source ./scripts/remove_blockage.tcl
```

```
source ./scripts/antenna_route_guide.tcl
```

```
source ./scripts/save_cell.tcl
```

```
geSaveAs
```

```
setFormField "Save As" "Cell Name" "03_pre_cts"
```

```
formOK "Save As"
```

## 1.6 クロックツリー生成

CTS(Clock Tree Synthesis) を舐めてはいけない。sdc ファイルがちゃんとしてないと中々言うことを聞いてくれない。POCOP.tcl では組み合わせ回路ではもちろん、これは不要なのでスクリプトで判断して、以下を実行している。

まずは準備。

```
source ./scripts/pre_cts_opt.tcl
```

```
-----中身-----
```

```
astAutoPlace
```

```
formDefault auto_place
```

```
setFormField auto_place effort Medium
```

```
setFormField auto_place pre_place_optimization 0
```

```
setFormField auto_place in_place_optimization 0
```

```

setFormField auto_place post_place_optimization 1
setFormField auto_place post_cts_optimization 0
setFormField auto_place fix_hold 1
formOK auto_place

astSearchRefine
formDefault place_cells_-_search_&_refine
setFormField place_cells_-_search_&_refine speed_per_area: medium
formOK place_cells_-_search_&_refine
formCancel place_cells_-_search_&_refine

```

```

pdsCROptimization
astTranFix
astCapFix

```

クロックツリーの生成。使えるバッファ、遅延を設定する。

```

source ./scripts/cts.tcl
-----中身-----
astClockOptions
formDefault clock_common_options
setFormField clock_common_options clock_nets "CLK"
setFormField clock_common_options buffers/inverters "SC23CKBUFCLXL1 SC23CKBUFCLXP1 SC23CKBUFCLXR1 SC23CKBUFCLXH1"
setFormField clock_common_options buffer_sizing:_leq_cells "SC23CKBUFCLXH1 SC23CKBUFCLXL1 SC23CKINVBCLXH1 SC23CKINVBCLXL1"
setFormField clock_common_options delay_cells "SC23DLY02XC1 SC23DLY02XH1 SC23DLY04XC1 SC23DLY04XH1"
setFormField clock_common_options synthesis_effort "5"
setFormField clock_common_options optimization_effort "5"
setFormField clock_common_options ignore_sdc "0"
setFormField "Clock Common Options" "Synthesis Effort" "10"
setFormField "Clock Common Options" "Optimization Effort" "10"
formButton "Clock Common Options" "ConstraintSubForm"
setFormField "Clock Common Options" "Maximum Fanout" "10"
setFormField "Clock Common Options" "Best" "1"
subFormHide "Clock Common Options" "3"
formOK clock_common_options

```

```

astHFCTS
setFormField "High Fanout Clock Tree Synthesis" "Fix DRC for all Unsynthesized Clock Nets" "No"
setFormField "High Fanout Clock Tree Synthesis" "Net File Name" ""
formOK "High Fanout Clock Tree Synthesis"
astCTO
formDefault clock_tree_optimization
formOK clock_tree_optimization

```

クロックツリーを表示してみる。ここで、ちゃんとできているかをチェックする。水色でツリーができているのがわかる。

```

source ./scripts/show_clk_tree.tcl

```

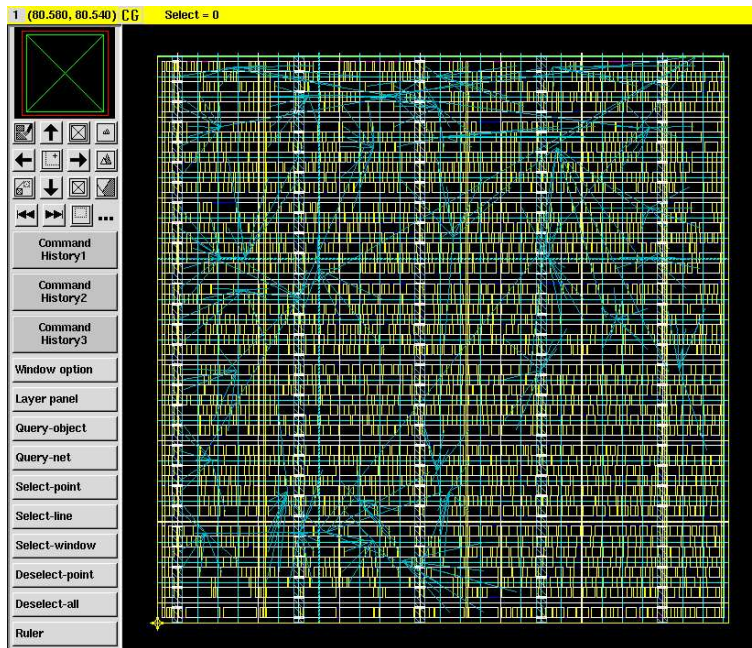


図 5: クロックツリー

次にホールドタイムの調整、その後保存。

```
source ./scripts/fix_hold.tcl
source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "05_pre_post_cts_place_opt"
formOK "Save As"
```

再び、最適化、再びホールドタイム調整

```
source ./scripts/post_cts_timing_setup.tcl
source ./scripts/post_cts_place_opt.tcl
source ./scripts/fix_hold.tcl
source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "06_pre_insert_cuba"
formOK "Save As"
```

## 1.7 Cubaを入れる

YUZCUBA はフィラーセルといってセル間のすき間を埋めるためのセルの一種であるが、容量と金属が入っている。これを一定間隔でいれてやらないと ERC が発生してしまう。そこで、まずこれを入れられる所に全面に入れて、後で必要以外を消去するという方針で挿入する。この辺は社長／いけぶちさんのオリジナル技と言っていいと思う。

```
source ./scripts/insert_cuba.tcl
-----中身-----
set lx 1.1
set ly [expr 1.2 - 0.16]
set rx [expr $lx + $core_width]
```

```

set ry [expr $ly + $score_height]

set interval_x 19.8
set interval_y 19.8

# fill all empty area by CUBA
axgAddFillerCellByArea
formDefault add_filler_cell_by_area
setFormField add_filler_cell_by_area master_cell_name(s)_with_metal "SC23YUZCUBAS081"
setFormField add_filler_cell_by_area connect_to_power_net_(optional) VDD
setFormField add_filler_cell_by_area connect_to_ground_net_(optional) VSS
formHide add_filler_cell_by_area

addPoint 1 [concat $lx $ly]
addPoint 1 [concat $rx $ry]
abortCommand

# select
gePointSelect
for {set y $ly} {$y < $ry} {set y [expr $y + $interval_y]} {
    for {set x $lx} {$x < $rx} {set x [expr $x + $interval_x]} {
addPoint 1 [concat [expr $x + 0.6] [expr $y + 0.9 + 0.16]]
    }
}
abortCommand

# fix
aprCmdFixCell
setFormField set_fixed_status mark_object fixed
formOK set_fixed_status

# remove extra filler
axgPurgeFillerCell
setToggleField purge_filler purge_type std_cell 1
setToggleField purge_filler purge_type pad 0
setToggleField purge_filler purge_type tap 0
formOK purge_filler

geDeselectAll

で、いつものように、電源、グラウンドの接続をして保存する。

source ./scripts/connect_pg.tcl

source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "07_pre_auto_route"

```

```
formOK "Save As"
```

## 1.8 ルーティング

最初にホールドタイム調整をしておく。

```
source ./scripts/fix_hold.tcl
```

次にアンテナのルールを読み込む。これはスキームで書かれているのでスキーム入力にして読ませる。富士通にお願いしてもらったもの。これでアンテナが無くなると信じていたのだが、考えが甘かった。でも、もちろんやらないよりはましなので、必ず読み込ませる。

```
load_scheme ./scripts/antenna_rule.scm
```

それで、ルーティングのセットアップをする。12層を使うと Astro が落ちるので、11層以下にする。MET1は使わないので、11層から2層を利用する。

```
source ./scripts/route_setup.tcl
```

```
axgSetRouteOptions
```

```
formDefault route_common_options
```

```
setFormField route_common_options timing_driven 1
setFormField route_common_options track_assign_timing_driven 0
setFormField route_common_options same_net_notch "check and fix"
setFormField route_common_options fat_wire_checking "merge then check"
setFormField route_common_options detail_route_timing_driven 1
setFormField route_common_options crosstalk_prevention 1
setFormField route_common_options single-row/column_via_array optimize
setFormField route_common_options stack_via_less_than_min-area forbid
setFormField route_common_options skew_control 1
formOK route_common_options
```

```
axgSetMinMaxLayer
```

```
setFormField set_min_max_layer max_metal_routing_layer 11
setFormField set_min_max_layer min_cell_layer MET2
setFormField set_min_max_layer max_cell_layer METG2
setFormField set_min_max_layer min_layer MET2
setFormField set_min_max_layer max_layer METG2
formOK set_min_max_layer
```

まずはクロックのルーティングを行う。

```
source ./scripts/route_clk.tcl
```

次にルーティングをする。10回はいいい加減に決めた。実行 CPU 数は実行環境に応じて設定のこと。ここは時間が掛かる。

```
source ./scripts/auto_route.tcl
```

```
-----中身-----
```



```

axgAutoRoute
setFormField auto_route global_route_speed medium
setFormField auto_route search_&_repair_loop 10
setToggleField auto_route save_after_phase global 1
setToggleField auto_route save_after_phase detail 1
# setFormField "Auto Route" "Distributed Routing" "1"
# setFormField "Auto Route" "Number of CPUs" "2"
formOK auto_route

```

結果は保存しておく。

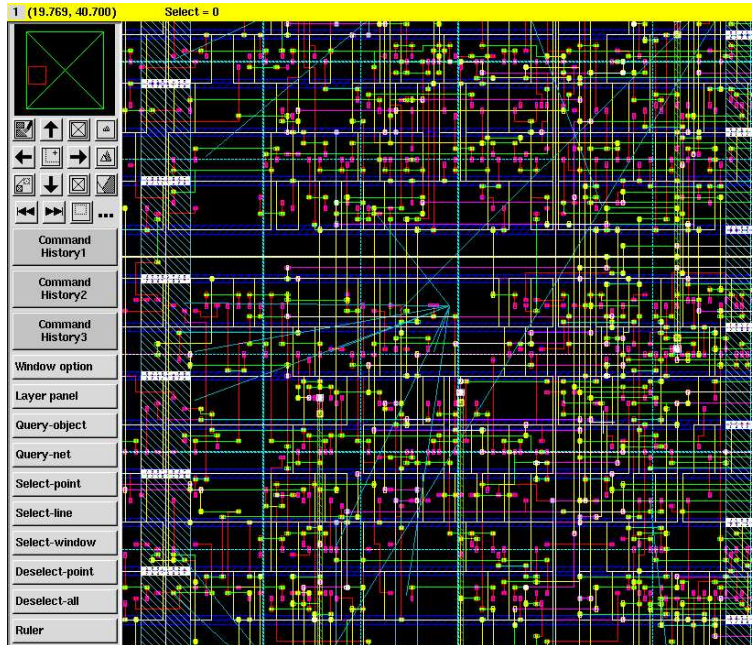


図 6: ルーティング後

```

source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "08_post_route"
formOK "Save As"

```

## 1.9 最適化

以下は最適化である。コマンドをほとんどそのまま適用しているので、中身は多分示さなくても良いかと思う。  
最初にクロックツリーの最適化を行う。

```

source ./scripts/post_route_timing_setup.tcl
if {!$is_combinational_circuit} {
source ./scripts/post_route_cto.tcl
source ./scripts/remove_clk_antenna.tcl
}

source ./scripts/save_cell.tcl

```

```
geSaveAs
setFormField "Save As" "Cell Name" "09_pre_post_route_opt"
formOK "Save As"
```

次に全体の最適化を行う。search and repair の数等は適当に設定してある。

```
source ./scripts/post_route_opt.tcl
source ./scripts/connect_pg.tcl
source ./scripts/search_repair.tcl
source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "10_pre_insert_filler"
formOK "Save As"
```

最後に仕上げである。

```
source ./scripts/fix_hold.tcl
source ./scripts/via_opt.tcl
source ./scripts/remove_blockage.tcl
source ./scripts/remove_route_guide.tcl
source ./scripts/insert_filler.tcl
source ./scripts/fill_notch.tcl
source ./scripts/connect_pg.tcl
source ./scripts/save_cell.tcl
geSaveAs
setFormField "Save As" "Cell Name" "11_final"
formOK "Save As"
```

ここで、重要なのは、filler を入れる部分である。

```
source ./scripts/insert_filler.tcl
-----中身-----
axgAddFillerCell
formDefault add_filler_cell

setFormField add_filler_cell master_cell_name(s) SC23YUZS021,SC23YUZS011

setFormField "Add Filler Cell" "respect overlap check objects" "1"
setFormField add_filler_cell connect_to_power_net_(optional) VDD
setFormField add_filler_cell connect_to_ground_net_(optional) VSS
setFormField add_filler_cell between_std_cells_only 0
formOK add_filler_cell
```

ここでは、Cuba が一定間隔で入っていることを前提に YUZS で細かい所を埋めている。

注意：ここで YUZB を入れないと密度チェックに引っかかる可能性がある。

最終的にレイアウトが全て埋まっていることを確認する。

これで安心してはダメで、

```
source scripts/create_macro.tcl
```



図 7: 最終レイアウト

を実行する。これはマクロを作るもので、これをやらないと上位のレイアウトから使うことができない。

```
source scripts/verilog_out_last.tcl
source scripts/stream_out.tcl
```

これで、前者は、lvs を行うための、POCOP\_lvs.v が、後者で drc を行うための、POCOP.gds が、 postlayout\_net に吐き出される。

ちなみに、ここで、Verify をクリックし、DRC, LVS などをやっておくのも手だが、これは大抵上手く行く。ここで通らなかったらかなり事態は深刻なので、必ず修正してエラーフリーにしておく必要がある。しかし、本当に問題なのはラスボスである calibre で通るか、ということだ。

ふんがは、この段階で必ず calibre の DRC をやっておくことをお勧めする。というのは、ここで出れば全体でもでるに決ってるからだ。LVS もやった方がいいのだが、出力ピンファイルをつくるのが面倒で、どっちみち後で上位からやれば良いので、ここでは LVS は避けても良いと思う。ANT と ERC も最後にやれば良いと思うが、ケースバイケースである。やり方は、calibre ラスボス 4 人衆のマニュアルを参照されたい。

さて、これでマクロはできたのだが、ここで一つ言っておきたいのは、マクロができたと喜んでいてはならない。パッドや電源を含んだチップを通すのは、たとえマクロ以外は空っぽでも、マクロを作った 3 倍以上の時間とテクが必要である。これは POCOP\_TOP を作るマニュアルを参照されたい。