DIC_HW4

學號：411086006 姓名：張愷和

一、 DCT 更改輸入輸出

這次程式碼架構可以想成，testbench 使用一個 state 來控制 module 執行的狀況。當 state 等於 0 的時候，將 in 的資料一個一個輸入進 in_mem 裡，最多 8 個，state 等於 1 的時候，將 in_mem 裡的資料拿出來計算 DCT，state 等於 2 的時候，將計算結果輸入進 out_mem，state 等於 3 的時候，將計算結果從 out_mem 一個一個輸出到 out，out_mem 最多儲存量為 8。

```
for (i = 0; i <= 64; i = i + 1) begin
    if(i % 8 == 0)
    begin
        state = 1;
        #10;
        state = 2;
        #50;
        rstn = 0;
        state = 3;
        #100;
    end
    rstn = 1;
    state = 0;
    in = mem0[i];
    #10;
end
```

圖一：testbench 使用 state 來控制輸入及輸出，rstn 可以重製 in_mem 的空間。

```
always @(posedge clk) begin
    if(rstn == 0) begin
        counter1 <= 0;
    end else if(state == 0)begin
        in_mem[counter1] <= in;
        counter1 <= counter1 + 1;
    end else begin
        in_mem[counter1] <= in_mem[counter1];
        counter1 <= counter1;
    end
end
```

圖二：state = 0，輸入最多 8 個數據

```
always @(posedge clk) begin
    if(state == 1) begin
        s10 <= in_mem[0] + in_mem[7];
        s17 <= in_mem[0] - in_mem[7];
        s11 <= in_mem[1] + in_mem[6];
        s16 <= in_mem[1] - in_mem[6];
        s12 <= in_mem[2] + in_mem[5];
        s15 <= in_mem[2] - in_mem[5];
        s13 <= in_mem[3] + in_mem[4];
        s14 <= in_mem[3] - in_mem[4];
    end
end
```

圖三：state = 1，開始計算 1D-DCT

```
always @(posedge clk) begin
    if(state == 2) begin
        out_mem[0] <= o0;
        out_mem[1] <= o1;
        out_mem[2] <= o2;
        out_mem[3] <= o3;
        out_mem[4] <= o4;
        out_mem[5] <= o5;
        out_mem[6] <= o6;
        out_mem[7] <= o7;
    end
end
```

圖四：state = 2，計算結果輸入到 out_mem

```
always @(posedge clk) begin
    if(state == 3) begin
        out <= out_mem[counter4];
        counter4 <= counter4 + 1;
    end else begin
        counter4 <= 0;
    end
end
```

圖五：state = 3，將 out_mem 結果輸出到 out

會這樣設計程式碼是因為之前有使用過[8:0] in_mem [0:64]和[12:0] out_mem [0:64]的組合，但是在合成的時候面積和時間都沒有很理想，因此最終改成上面那種版本。

功能測試：

/stimulus_1ddct/out	12'd924	532	417	-57	-180	96	197	62	-71
/stimulus_1ddct/out	12'd149	924	510	-6	-228	200	63	13	-60
/stimulus_1ddct/out	12'd31	1193	149	-81	-49	-5	9	38	-61
/stimulus_1ddct/out	12'd33	1182	31	-16	-29	-18	38	-16	-15
/stimulus_1ddct/out	12'd1095	1203	33	59	-30	37	-12	19	-15
/stimulus_1ddct/out	-12'd25	1095	-18	7	39	35	-12	-31	-6
/stimulus_1ddct/out	12'd417	1110	11	-47	-25	44	28	3	-45
/stimulus_1ddct/out	12'd417	1192	77	-54	-3	38	49	22	31

圖六：使用教授提供範例 txt 檔，使用單輸入單輸出計算 DCT

```
1D DCT
  532.000000  924.000000 1193.000000 1182.000000 1203.000000 1095.000000 1110.000000 1192.000000
  419.597402  511.751725  150.707195   33.453284   34.528970  -16.663525   13.880460   79.638702
  -56.438241   -5.411961  -80.311861  -15.044705   59.112359    7.986625  -46.008267  -53.585013
 -178.533266 -226.857421  -47.913577  -28.179420  -29.743516   39.677172  -24.164651   -2.221957
   96.000000  200.000000   -5.000000  -18.000000   37.000000   35.000000   44.000000   38.000000
  198.084961   62.847978   10.350396   38.829415  -11.284208  -12.145579   27.165193   50.401941
   62.343604   13.065630   38.678223  -15.416123   19.892939  -30.367974    3.903758   22.195639
  -71.401970  -59.598779  -60.995901  -13.677523  -14.688095   -7.248026  -44.152630   30.990070
```

圖七：使用教授提供 c 語言 DCT 計算程式碼，計算理想 DCT 數值

比對圖六和圖七，各項數值的差距很小，範圍幾乎都在±2 之間，看起來計算功能上應該沒有問題。

二、 Synthesis

接著開始合成，因為等一下 APR 需要設定 I/O pad，所以需要先將計算 DCT 的 module 和一個可以設定輸入輸出的檔案合成在一起，教授在 lab 是使用一個 CHIP.v 的檔案來設定，這邊就直接修改那個檔案來設定輸入跟輸出。

```
loeffler_1d CORE ( .clk(i_clk), .rstn(i_rstn), .in(i_in), .out(i_out), .state(i_state));

XMD ipad_clk    ( .I(clk), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_clk) );
XMD ipad_rst    ( .I(rstn), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_rstn) );
XMD ipad_state0 ( .I(state[0]),  .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_state[0]));
XMD ipad_state1 ( .I(state[1]),  .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_state[1]));

XMD ipad_in0    ( .I(in[0]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[0]) );
XMD ipad_in1    ( .I(in[1]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[1]) );
XMD ipad_in2    ( .I(in[2]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[2]) );
XMD ipad_in3    ( .I(in[3]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[3]) );
XMD ipad_in4    ( .I(in[4]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[4]) );
XMD ipad_in5    ( .I(in[5]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[5]) );
XMD ipad_in6    ( .I(in[6]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[6]) );
XMD ipad_in7    ( .I(in[7]), .PU(n_Logic0_), .PD(n_Logic0_), .SMT(n_Logic0_), .O(i_in[7]) );

YA2GSD opad_out0  ( .I(i_out[0]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[0]) );
YA2GSD opad_out1  ( .I(i_out[1]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[1]) );
YA2GSD opad_out2  ( .I(i_out[2]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[2]) );
YA2GSD opad_out3  ( .I(i_out[3]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[3]) );
YA2GSD opad_out4  ( .I(i_out[4]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[4]) );
YA2GSD opad_out5  ( .I(i_out[5]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[5]) );
YA2GSD opad_out6  ( .I(i_out[6]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[6]) );
YA2GSD opad_out7  ( .I(i_out[7]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[7]) );
YA2GSD opad_out8  ( .I(i_out[8]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[8]) );
YA2GSD opad_out9  ( .I(i_out[9]),  .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[9]) );
YA2GSD opad_out10 ( .I(i_out[10]), .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[10]) );
YA2GSD opad_out11 ( .I(i_out[11]), .E(n_Logic1_), .E2(n_Logic1_), .E4(n_Logic1_), .E8(n_Logic1_), .SR(n_Logic1_), .O(out[11]) );
```

圖八：設定輸入和輸出位，XMD 表示輸入，YA2GSD 表示輸出。

從圖八可以看到，輸入使用 XMD module 設定，輸出使用 YA2GSD

module 設定，由於每個只能設定 1 個 bit，所以看起來會有很多行，以 out 為例，out 有 12 個 bits，所以 YA2GSD 的 opad 需要設定 12 個 bits。

然後就將這個檔案和 loeffler_1d.v 的檔案合成再一起看 time、power 以及 area report。

```
****************************************
Report : area
Design : CHIP
Version: R-2020.09
Date   : Tue May 28 15:45:00 2024
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    fsa0m_a_generic_core_ss1p62v125c (File: /home/cad/tech/CBDK018_U
    fsa0m_a_t33_generic_io_ss1p62v125c (File: /home/cad/tech/CBDK018_

Number of ports:                         1293
Number of nets:                          4415
Number of cells:                         2525
Number of combinational cells:           1919
Number of sequential cells:               545
Number of macros/black boxes:              24
Number of buf/inv:                        498
Number of references:                       5

Combinational area:              60877.354074
Buf/Inv area:                     4318.473568
Noncombinational area:           29735.597179
Macro/Black Box area:           210583.546875
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                301196.498128
Total area:                 undefined
```

圖九：area report, total cell area=301196.498128

```
clock clk (rise edge)                                  10.00      10.00
clock network delay (ideal)                             1.00      11.00
clock uncertainty                                      -0.10      10.90
CORE/s26_reg_11_/CK (QDFFN)                             0.00      10.90 r
library setup time                                     -0.14      10.76
data required time                                                10.76
---------------------------------------------------------------------------
data required time                                                10.76
data arrival time                                                -10.76
---------------------------------------------------------------------------
slack (MET)                                                       0.00
```

圖十：time report, slack time=0.00

Power Group	Internal Power	Switching Power	Leakage Power	Total Power	(%)	Attrs
io_pad	3.0032	8.9337e-02	1.4811e+06	3.0941	(54.96%)	
memory	0.0000	0.0000	0.0000	0.0000	(0.00%)	
black_box	0.0000	0.0000	0.0000	0.0000	(0.00%)	
clock_network	0.0000	0.0000	0.0000	0.0000	(0.00%)	
register	2.2208	3.4040e-02	2.0052e+06	2.2568	(40.09%)	
sequential	0.0000	0.0000	0.0000	0.0000	(0.00%)	
combinational	9.7272e-02	0.1788	2.9943e+06	0.2791	(4.96%)	
Total	5.3213 mW	0.3022 mW	6.4806e+06 pW	5.6300 mW		

圖十一：power report, total power=5.6300mW

三、 Presim

使用剛剛合成的 CHIP_syn 檔案透過 vsim 來生成 vcd 檔案，然後使用 primetime 來計算功率和時間。

```
vsim -64 -t ns -sdfnowarn -vopt -sdftyp /dct_1d=/home2/VLSI003/yourHomeWork/HW4/syn/mapped_18/CHIP_syn.sdf -multisource_de

# vcd is value change dump
vcd file $design.vcd
vcd add -r $tp_instance/dct_1d/*


run -all
```

圖十二：tcl 設定 vsim 模擬加入 mapped CHIP_syn.sdf 檔案生成 vcd 檔案

這次 primetime 設定中需要加入 syn link library 檔案，如果使用之前的 tcl 檔案會跑不過 power 計算，然後計算 power 和 time 需要加入 sdc 和 spef 檔案來計算。

```
set search_path          "$mapped_path $tech_lib_path . "
set target_library     "fsa0m_a_generic_core_ss1p62v125c.db fsa0m_a_generic_core_ff1p98vm40c.db fsa0m_a_generi
set link_library       "* $target_library"

read_verilog         $mapped_path/CHIP_syn.v
current_design       $design
link


########################################################################
#       set transition time / annotate parasitics
########################################################################
read_sdc $mapped_path/CHIP_syn.sdc
#set_disable_timing [get_lib_pins ssc_core_typ/*/G]
read_parasitics $mapped_path/CHIP_syn.spef
```

圖十三：tcl 更改 link_library target library 和加入 CHIP_syn sdc spef 檔案

最後計算出 time report 和 power report。

```
                 Internal  Switching  Leakage    Total
Power Group      Power     Power      Power      Power   (     %)  Attrs
--------------------------------------------------------------------------------
clock_network    2.177e-03    0.0000    0.0000 2.177e-03 (77.20%)  i
register         1.000e-04 1.863e-04 2.011e-06 2.883e-04 (10.22%)
combinational    1.562e-04 1.955e-04 2.990e-06 3.546e-04 (12.58%)
sequential          0.0000    0.0000    0.0000    0.0000 ( 0.00%)
memory              0.0000    0.0000    0.0000    0.0000 ( 0.00%)
io_pad              0.0000    0.0000    0.0000    0.0000 ( 0.00%)
black_box           0.0000    0.0000    0.0000    0.0000 ( 0.00%)

  Net Switching Power          = 3.818e-04   (13.54%)
  Cell Internal Power          = 2.433e-03   (86.29%)
  Cell Leakage Power           = 5.001e-06   ( 0.18%)
                                 ---------
Total Power                    = 2.820e-03  (100.00%)

X Transition Power             = 7.607e-06
CAPP Estimated Glitching Power =    0.0000

Peak Power                     = 6.242e-03
Peak Time                      =      1450
```

圖十四：primetime power report, total power=2.8mW

```
clock clk (rise edge)                    10.00      10.00
clock network delay (propagated)          0.00      10.00
clock reconvergence pessimism             0.00      10.00
clock uncertainty                        -0.10       9.90
s32_reg_11_/CK (QDFFN)                               9.90 r
library setup time                       -0.13       9.77
data required time                                   9.77
--------------------------------------------------------------
data required time                                   9.77
data arrival time                                   -9.65
--------------------------------------------------------------
slack (MET)                                          0.11
```

圖十五：primetime time report, slack time=0.11

以 presim 結果上來看，程式上看起來沒有 time violation 的問題，可以正常使用。

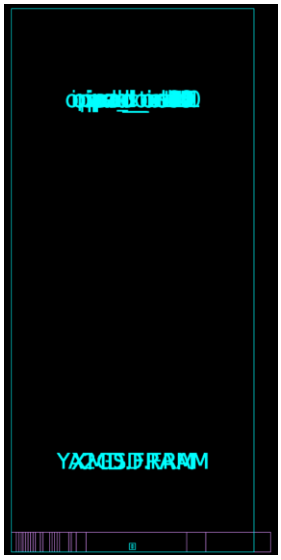
四、 APR

現在使用剛剛合成的 CHIP_syn 檔案來做 APR，首先需要把 tdf 檔案設定好輸入、輸出、core vcc、core gnd、io vcc 和 io gnd 都安排好。

```
set_pad_physical_constraints -side 1 -pad_name io_vcc1      -order 1    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_rstn        -order 2    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name io_gnd1      -order 3    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_in0     -order 4    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_in1     -order 5    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name core_vcc1    -order 6    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_in2         -order 7    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name core_gnd1    -order 8    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_in3         -order 9    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_in4         -order 10   -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 1 -pad_name ipad_in5         -order 11   -min_left_iospace 31 -min_right_iospace 31


set_pad_physical_constraints -side 2 -pad_name io_vcc2      -order 1    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name ipad_in6         -order 2    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name ipad_in7         -order 3    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name io_gnd2      -order 4    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name ipad_state0  -order 5    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name core_vcc2    -order 6    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name ipad_clk     -order 7    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name opad_out0        -order 8    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name io_vcc3      -order 9    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name core_gnd2    -order 10   -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 2 -pad_name io_gnd3      -order 11   -min_left_iospace 31 -min_right_iospace 31


set_pad_physical_constraints -side 3 -pad_name opad_out1        -order 1    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name opad_out2        -order 2    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name core_gnd4        -order 3    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name opad_out3        -order 4    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name opad_out4    -order 5    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name core_vcc4        -order 6    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name opad_out5    -order 7    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name opad_out6        -order 8    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name io_gnd4      -order 9    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name opad_out7            -order 10   -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 3 -pad_name io_vcc4      -order 11   -min_left_iospace 31 -min_right_iospace 31


set_pad_physical_constraints -side 4 -pad_name io_vcc5          -order 1    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name opad_out8            -order 2    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name io_gnd5          -order 3    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name core_gnd3            -order 4    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name opad_out9            -order 5    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name core_vcc3            -order 6    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name opad_out10           -order 7    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name io_vcc6      -order 8    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name opad_out11       -order 9    -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name ipad_state1      -order 10   -min_left_iospace 31 -min_right_iospace 31
set_pad_physical_constraints -side 4 -pad_name io_gnd6      -order 11   -min_left_iospace 31 -min_right_iospace 31
```

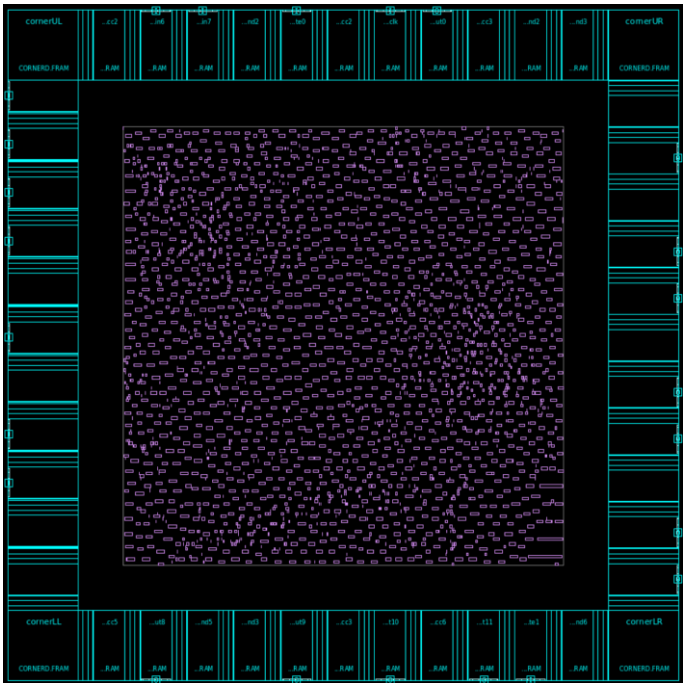
圖十六：設定輸入輸出 vcc 和 gnd 腳位，vcc 的位置盡量都放在中間。

接著就可以開始跑 tcl 檔案，一開始先 create design 並且 import 剛剛合成的 CHIP_syn 檔案。



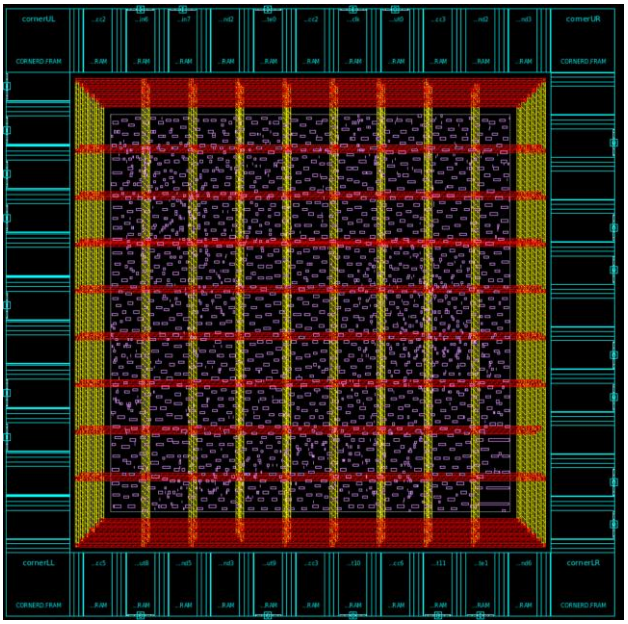
圖十七：create design

然後引入 tdf 檔案安排腳位做 I/O ring，I/O ring 設定完後需要用 insert_pad_filler 指令將 pad 之間的空隙填起來，之後再 create fp_placement 把 standrad cells 放到 CHIP 裡面。



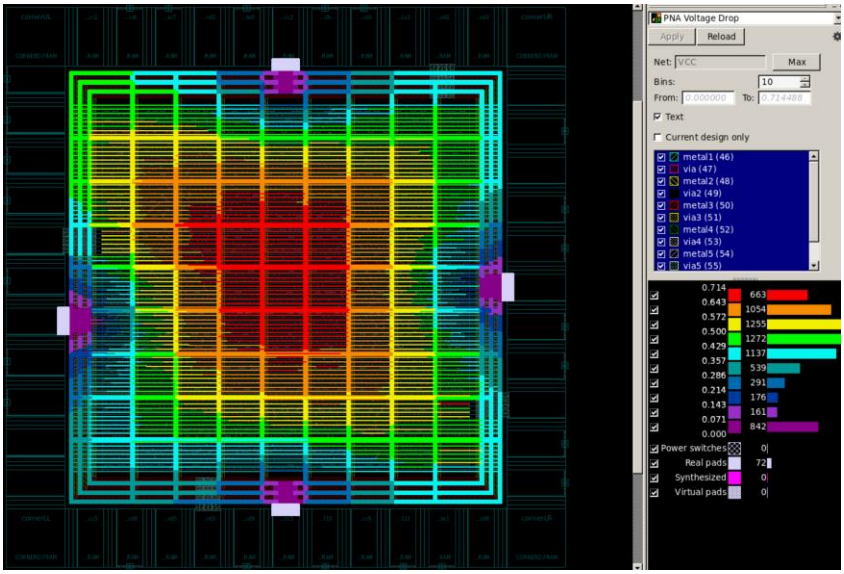
圖十八：設定好 I/O ring 和把 cell 放到 CHIP 裡面。

接著設定 fp_rail，設置電源和地線軌道。這些軌道確保了芯片中各個部分都能夠獲得穩定的電源供應和接地。



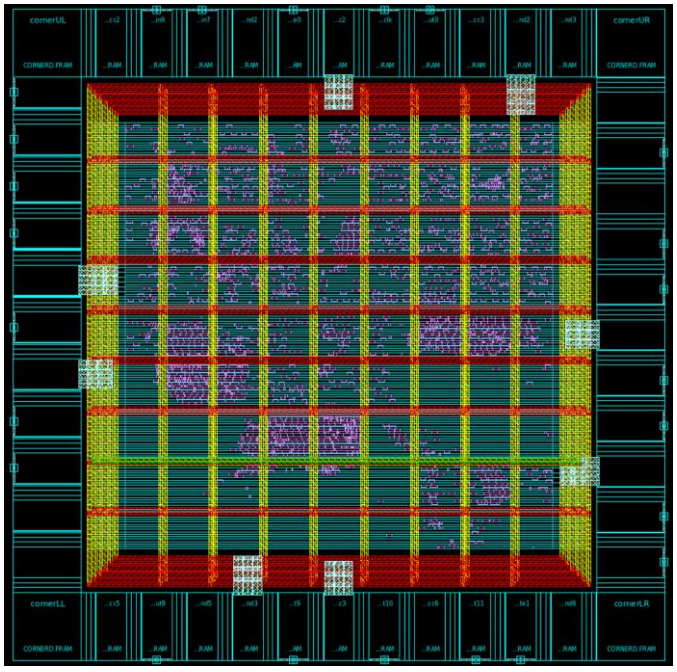
圖十九：create fp_rail

再來需要分析 fp_rail 供電的情況，可以確認電源輸入到所有負載的電壓降是否在可接受範圍內，盡量讓圖上的紅色區域越小會越好。作業實作上，目前擺放的方式紅色區域已經是最小的了，如果要在更小可能要加一二個 core vcc 和 gnd 來供電。



圖二十：analysis fp_rail

設定 clock tree，目的是設計和生成一個均勻分佈的時鐘樹，將時鐘信號分配到所有時鐘端點，確保時鐘信號的延遲和偏差（skew）在可接受的範圍內。

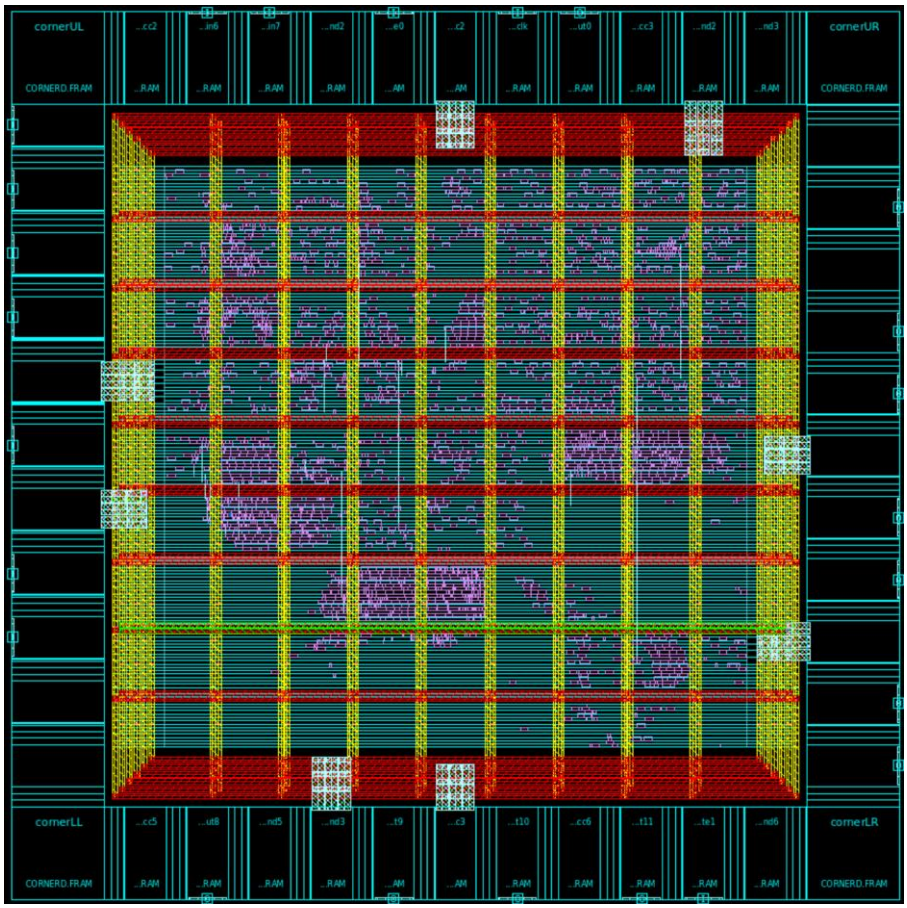


圖二十一：clock tree synthesis

```
clock clk (rise edge)                                  10.00      10.00
clock network delay (propagated)                        1.31      11.31
clock uncertainty                                      -0.10      11.21
CORE/s32_reg_11_/CK (QDFFN)                             0.00      11.21 r
library setup time                                     -0.14      11.07
data required time                                                11.07
-------------------------------------------------------------------------
data required time                                                11.07
data arrival time                                                -10.97
-------------------------------------------------------------------------
slack (MET)                                                       0.10
```

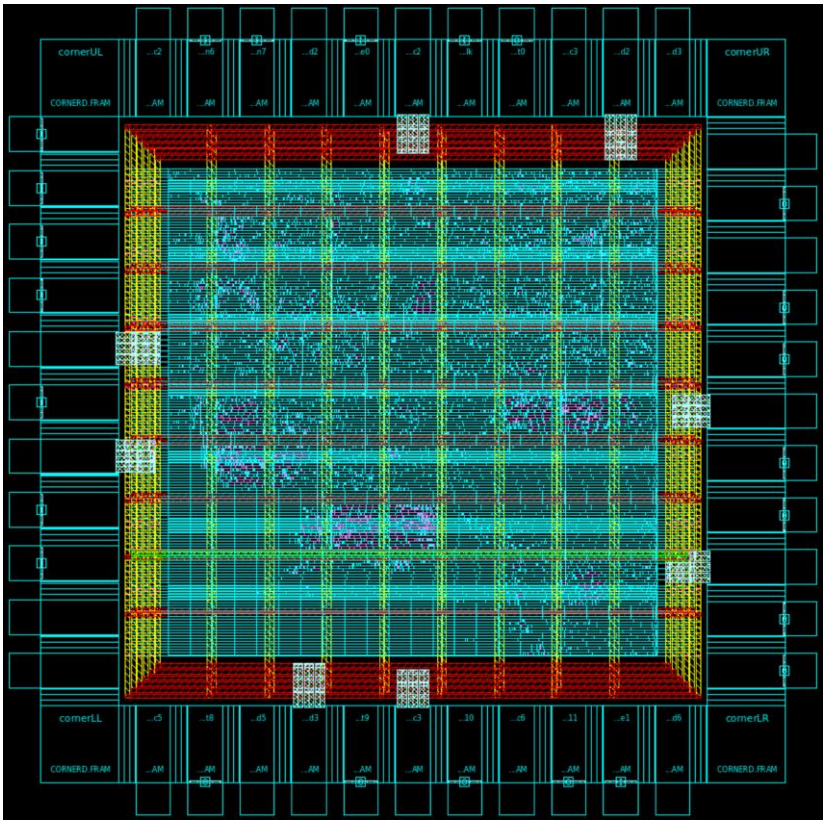
圖二十二：clock tree timing report

Routing，在電路元件之間的連接線路設置路徑，在圖二十三中可以看到紫色 cell 的區域會有更淡藍色的線連在一起。



圖二十三：route

最後就 DFM，主要目的要做 core filler 和 io bondpad，core 就是把 CHIP 中間的部分填滿，跟 pad filler 滿像的，io bondpad 就是需要設定引腳，然後這些做完就可以輸出 report 和 postsim 需要的檔案。



圖二十四：cell filler and io bondpad

```
****************************************
Report : area
Design : CHIP
Version: R-2020.09
Date   : Tue May 28 22:47:25 2024
****************************************

Information: Updating design information... (UID-85)
Information: Input delay ('fall') on clock port 'clk' will be added to the clock's propagated skew. (TIM-112)
Information: Input delay ('rise') on clock port 'clk' will be added to the clock's propagated skew. (TIM-112)
Information: Input delay ('rise') on clock port 'clk' will be added to the clock's propagated skew. (TIM-112)
Information: Input delay ('fall') on clock port 'clk' will be added to the clock's propagated skew. (TIM-112)
Library(s) Used:

    fsa0m_a_t33_generic_io_ss1p62v125c (File: /home/cad/tech/CBDK018_UMC_Faraday_v1.1/CIC/SynopsysDC/db/fsa0m_a_t33_g
    fsa0m_a_generic_core_ss1p62v125c (File: /home/cad/tech/CBDK018_UMC_Faraday_v1.1/CIC/SynopsysDC/db/fsa0m_a_generic

Number of ports:                         24
Number of nets:                          56
Number of cells:                         43
Number of combinational cells:            6
Number of sequential cells:              12
Number of macros/black boxes:            24
Number of buf/inv:                        4
Number of references:                     9

Combinational area:              66758.227751
Buf/Inv area:                     6933.931139
Noncombinational area:           29894.961933
Macro/Black Box area:           210583.546875
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                307236.736559
Total area:                 undefined

Information: This design contains black box (unknown) components. (RPT-8)
```

圖二十五：area report, total cell area=307236.736559

```
Design        Wire Load Model           Library
------------------------------------------------
CHIP                  G5K               fsa0m_a_generic_core_ss1p62v125c


Global Operating Voltage = 1.62
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =   5.7210 mW   (82%)
  Net Switching Power  =   1.2509 mW   (18%)
                         ---------
Total Dynamic Power    =   6.9719 mW  (100%)

Cell Leakage Power     =   7.1269 uW


                 Internal         Switching           Leakage           Total
Power Group      Power            Power               Power             Power   (   %    ) Attrs
-----------------------------------------------------------------------------------------------
io_pad           3.1699           0.1220          1.4675e+06            3.2934  (  47.19%)
memory           0.0000           0.0000              0.0000            0.0000  (   0.00%)
black_box        0.0000           0.0000              0.0000            0.0000  (   0.00%)
clock_network    0.1513           0.9465          5.4956e+04            1.0979  (  15.73%)
register         2.2820        2.9686e-02         2.0204e+06            2.3137  (  33.15%)
sequential       0.0000           0.0000              0.0000            0.0000  (   0.00%)
combinational    0.1178           0.1526          3.5842e+06            0.2740  (   3.93%)
-----------------------------------------------------------------------------------------------
Total            5.7210 mW        1.2509 mW       7.1270e+06 pW         6.9790 mW
```

圖二十六：power report，total power = 6.9790mW

```
clock clk (rise edge)                                  10.00       10.00
clock network delay (propagated)                        1.27       11.27
clock uncertainty                                      -0.10       11.17
CORE/s32_reg_11_/CK (QDFFN)                             0.00       11.17 r
library setup time                                     -0.14       11.03
data required time                                                 11.03
-------------------------------------------------------------------------
data required time                                                 11.03
data arrival time                                                 -10.90
-------------------------------------------------------------------------
slack (MET)                                                         0.13
```

圖二十七：timing report，slack time=0.13

五、 Postsim

將剛剛 APR 完的檔案使用 vsim 加入 CHIP_route 檔案執行一遍，看功能有沒有錯誤，並且順便生成 vcd 檔案。

```
set design loeffler_1d
set source_path_pr  "source"
set mapped_path_pr  "../pr/mapped_18_pr"
set tech_lib /home/cad/tech/CBDK018_UMC_Faraday_v1.1/CIC/Verilog/fsa0m_a_generic_core_21.lib.src

set work_path work
set tp_module loeffler_1d_test
set tp_instance stimulus_1ddct

#set num_inputs 512

vlib $work_path
vlog $tech_lib -incr
vlog $mapped_path_pr/CHIP_route.v
#vcom $source_path_pr/$tp_module.vhd
vlog $source_path_pr/$tp_module.v

#vsim -c -l vsim.log -multisource_delay max -sdftyp /$tp_instance/=$mapped_path_pr/$design.sdf -sdfn

#vsim -voptargs=+acc -t ns -sdfnowarn -vopt -sdftyp /uut=/home/ccsun/Project/Lab3/syn/mapped_18/alu.

#vsim -voptargs=+acc -t ns -sdfnowarn -vopt -sdftyp /$tp_instance=$mapped_path_pr/$design.sdf $work_
vsim -64 -t ns -sdfnowarn -vopt -sdftyp /dct_1d=/home2/VLSI003/yourHomeWork/HW4/pr/mapped_18_pr/CHIP_

# vcd is value change dump
vcd file $design.vcd
vcd add -r $tp_instance/dct_1d/*


run -all
#run 1000 ns
```

圖二十八：vsim 執行加入 CHIP_route.sdf 檔案功能測試，並且記錄 vcd

/stimulus_1ddct/out	12'dX	532	417	-57	-180	96	197	62	-71
/stimulus_1ddct/out	12'dX	924	510	-6	-228	200	63	13	-60
/stimulus_1ddct/out	12'dX	1193	149	-81	-49	-5	9	38	-61
/stimulus_1ddct/out	12'dX	1182	31	-16	-29	-18	38	-16	-15
/stimulus_1ddct/out	12'dX	1203	33	59	-30	37	-12	19	-15
/stimulus_1ddct/out	12'dX	1095	-18	7	39	35	-12	-31	-6
/stimulus_1ddct/out	12'dX	1110	11	-47	-25	44	28	3	-45
/stimulus_1ddct/out	12'dX	1192	77	-54	-3	38	49	22	31

圖二十九：postsim 計算出來 1d-DCT 數值

```
1D DCT
  532.000000  924.000000 1193.000000 1182.000000 1203.000000 1095.000000 1110.000000 1192.000000
  419.597402  511.751725  150.707195   33.453284   34.528970  -16.663525   13.880460   79.638702
  -56.438241   -5.411961  -80.311861  -15.044705   59.112359    7.986625  -46.008267  -53.585013
 -178.533266 -226.857421  -47.913577  -28.179420  -29.743516   39.677172  -24.164651   -2.221957
   96.000000  200.000000   -5.000000  -18.000000   37.000000   35.000000   44.000000   38.000000
  198.084961   62.847978   10.350396   38.829415  -11.284208  -12.145579   27.165193   50.401941
   62.343604   13.065630   38.678223  -15.416123   19.892939  -30.367974    3.903758   22.195639
  -71.401970  -59.598779  -60.995901  -13.677523  -14.688095   -7.248026  -44.152630   30.990070
```

圖三十：使用教授提供 c 語言 DCT 計算程式碼，計算理想 DCT 數值

功能驗證上來看看起來沒有什麼問題。

最後有使用 primetime 來做 post power 的計算，不過在設定 tcl 過程中，一直出現錯誤，所以沒有計算出來實際的 time 和 power 是多少。