




# ASIC教材

## Chapter4 Innovus前置準備

高雄科技大學電子工程系  
B510實驗室

2021/1/29再編



# 目錄

為何需要使用Innovus ？

使用Design Vision合成DFT電路。

撰寫SDC File。

加入PAD到Gate-Level Verilog Code。

準備完成。



# 為何需要使用Innovus ？

自動繞線工具。

將設計實現成晶片。

Clock Timing分析與優化。



# 使用Design Vision合成DFT電路

DFT(Design For Testability)電路主要是可以讓測試儀器確認下線的IC電路是否有正常運作。

在使用Innovus之前利用Design Vision先合出DFT電路的 Gate-Level。

# 使用Design Vision合成DFT電路

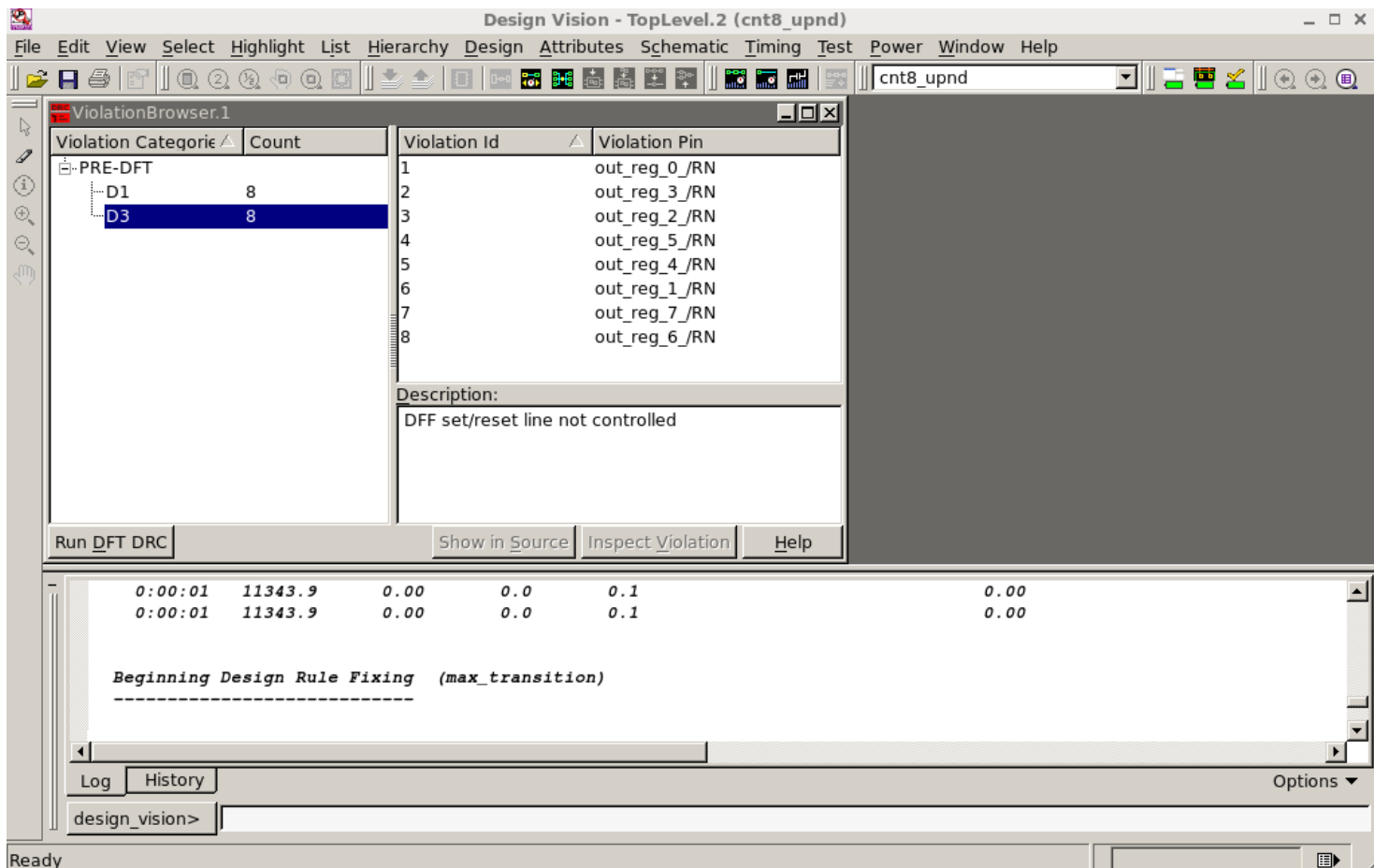
將之前第二章p.35所儲存的dc檔加入以下的指令，讀入dc檔之後，再照第二章p.37進行Compile。

```
create_port -dir in scan_en  
create_port -dir in scan_in  
create_port -dir out scan_out  
compile -scan -boundary_optimization  
set_dft_configuration -fix_set enable  
set_scan_configuration -chain_count 1  
set_scan_configuration -clock_mixing mix_edges
```

```
set_dft_signal -view exist -type ScanClock -timing {45 55} -port {?} 填clock的腳位名稱  
set_dft_signal -view exist -type Reset -active 1 -port ? 填reset的腳位名稱  
set_dft_signal -view spec -type ScanEnable -port scan_en -active 1  
set_dft_signal -view spec -type ScanDataIn -port scan_in  
set_dft_signal -view spec -type ScanDataOut -port scan_out  
create_test_protocol  
preview_dft -show all  
dft_drc  
insert_dft
```

# 使用Design Vision合成DFT電路

如果沒有填clock與reset會出現以下錯誤!!



# 使用Design Vision合成DFT電路

File > Save Design As 將含DFT電路的Gate-Level電路儲存，並改名稱叫「**XXX**\_gate\_dft.v」，前面章節未加DFT電路的Gate-Level

```
module cnt ( out, clk, rst );
    output [3:0] out;
    input clk, rst;
    wire n13, n14, n15, n16, N2, N3, N4, \add_8/carry[3] , \add_8/carry[2] ,
        n1, n4, n7, n8, n9, n11;

    JKFFRX4 out_reg_0_ ( .J(1'b1), .K(1'b1), .CK(clk), .RN(n1), .Q(n16), .QN(n4)
        );
    DFFRHQX4 out_reg_1_ ( .D(N2), .CK(clk), .RN(n1), .Q(n15) );
    DFFRHQX4 out_reg_2_ ( .D(N3), .CK(clk), .RN(n1), .Q(n14) );
    DFFRHQX4 out_reg_3_ ( .D(n7), .CK(clk), .RN(n1), .Q(n13) );
    XOR2X1 U5 ( .A(\add_8/carry[3] ), .B(out[3]), .Y(N4) );
    CLKINVSX2 U6 ( .A(n15), .Y(n9) );
    CLKINVSX2 U7 ( .A(n14), .Y(n11) );
    ADDHX4 U8 ( .A(out[2]), .B(\add_8/carry[2] ), .C0(\add_8/carry[3] ), .S(N3)
        );
    ADDHX4 U9 ( .A(out[1]), .B(n16), .C0(\add_8/carry[2] ), .S(N2) );
    CLKINVSX8 U10 ( .A(n4), .Y(out[0]) );
    CLKBUFX8 U11 ( .A(n13), .Y(out[3]) );
    CLKBUFX1 U12 ( .A(n8), .Y(n7) );
    CLKBUFXL U13 ( .A(N4), .Y(n8) );
    INVSX12 U14 ( .A(n9), .Y(out[1]) );
    INVSX12 U15 ( .A(n11), .Y(out[2]) );
    CLKINVSX8 U16 ( .A(rst), .Y(n1) );
endmodule
```

# 使用Design Vision合成DFT電路

## 加入DFT電路的Gate-Level

```
module cnt ( out, clk, rst, scan_en, scan_in, scan_out );
    output [3:0] out;
    input clk, rst, scan_en, scan_in;          加入scan腳位
    output scan_out;
    wire n33, n34, n35, N2, N3, N4, \add_8/carry[3], \add_8/carry[2], n10,
        n15, \out[3], n17, n28, n31, n37, n38;
    assign scan_out = \out[3];
    assign out[3] = \out[3];                  有DFT電路的DFF

    SDDFSRX2 out_reg_3_ ( .D(N4), .SI(n37), .SE(scan_en), .CK(clk), .SN(1'b1),
        .RN(n10), .Q(n28) );
    SDDFSRX2 out_reg_0_ ( .D(n31), .SI(scan_in), .SE(scan_en), .CK(clk), .SN(
        1'b1), .RN(n10), .Q(n35), .QN(n31) );
    SDDFSRX2 out_reg_2_ ( .D(N3), .SI(n38), .SE(scan_en), .CK(clk), .SN(1'b1),
        .RN(n10), .Q(n33), .QN(n37) );
    SDDFSRX2 out_reg_1_ ( .D(N2), .SI(n31), .SE(scan_en), .CK(clk), .SN(1'b1),
        .RN(n10), .Q(n34), .QN(n38) );
    CLKINVSX4 U12 ( .A(rst), .Y(n10) );
    BUFX12 U13 ( .A(n34), .Y(out[1]) );
    BUFX12 U14 ( .A(n33), .Y(out[2]) );
    ADDHX4 U15 ( .A(out[2]), .B(\add_8/carry[2]), .CO(\add_8/carry[3]), .S(N3)
        );
    ADDHX4 U16 ( .A(out[1]), .B(out[0]), .CO(\add_8/carry[2]), .S(N2) );
    CLKINVSX3 U17 ( .A(n28), .Y(n15) );
    CLKINVSX8 U18 ( .A(n15), .Y(\out[3]) );
    CLKINVSX3 U19 ( .A(n35), .Y(n17) );
    CLKINVSX8 U20 ( .A(n17), .Y(out[0]) );
    XOR2X1 U22 ( .A(\out[3]), .B(\add_8/carry[3]), .Y(N4) );
endmodule
```



# 加入PAD到Gate-Level Verilog Code

將之前Design Vision合出來的DFT Gate-Level另存新檔後，原本我們的電路設計只有9隻腳位，加上VSS、VDD、IOVSS、IOVDD 4隻腳位總共13隻腳位，晶片為四面所以補齊4倍數的腳位數再補上3隻空腳。

```
module cnt ( out, clk, rst, scan_en, scan_in, scan_out, nullpin );
    output [3:0] out;
    input clk, rst, scan_en, scan_in;      8
    input [2:0] nullpin; 3
    output scan_out; 1
    wire n33, n34, n35, N2, N3, N4, \add_8/carry[3], \add_8/carry[2], n10,
        n15, \out[3], n17, n28, n31, n37, n38;
    assign scan_out = \out[3];
    assign out[3] = \out[3];

    SDFFSRX2 out_reg_3 ( .D(N4), .SI(n37), .SE(scan_en), .CK(clk), .SN(1'b1),
        .RN(n10), .Q(n28) );
    SDFFSRX2 out_reg_0 ( .D(n31), .SI(scan_in), .SE(scan_en), .CK(clk), .SN(
        1'b1), .RN(n10), .Q(n35), .QN(n31) );
    SDFFSRX2 out_reg_2 ( .D(N3), .SI(n38), .SE(scan_en), .CK(clk), .SN(1'b1),
        .RN(n10), .Q(n33), .QN(n37) );
    SDFFSRX2 out_reg_1 ( .D(N2), .SI(n31), .SE(scan_en), .CK(clk), .SN(1'b1),
        .RN(n10), .Q(n34), .QN(n38) );
    CLKIN VX4 U12 ( .A(rst), .Y(n10) );
    BUFX12 U13 ( .A(n34), .Y(out[1]) );
    BUFX12 U14 ( .A(n33), .Y(out[2]) );
    ADDHX4 U15 ( .A(out[2]), .B(\add_8/carry[2]), .C0(\add_8/carry[3]), .S(N3)
        );
    ADDHX4 U16 ( .A(out[1]), .B(out[0]), .C0(\add_8/carry[2]), .S(N2) );
    CLKIN VX3 U17 ( .A(n28), .Y(n15) );
    CLKIN VX8 U18 ( .A(n15), .Y(\out[3]) );
    CLKIN VX3 U19 ( .A(n35), .Y(n17) );
    CLKIN VX8 U20 ( .A(n17), .Y(out[0]) );
    XOR2X1 U22 ( .A(\out[3]), .B(\add_8/carry[3]), .Y(N4) );
endmodule
```

# 加入PAD到Gate-Level Verilog Code

新增一個Module多加\_chip來區別這個Module是不是已經用PAD封裝，記得放置於最上面。

```
module cnt_chip ( pad_out, pad_clk, pad_rst, pad_scan_en, pad_scan_in, pad_scan_out ,pad_nullpin);
    output [3:0] pad_out;
    output pad_scan_out;
    input pad_clk, pad_rst, pad_scan_en, pad_scan_in;
    input [2:0] pad_nullpin;

    wire [3:0] out;
    wire clk, rst, scan_en, scan_in;
    wire [2:0] nullpin;
    wire scan_out;

    //Chip Kernel-----
    cnt k1( out, clk, rst, scan_en, scan_in, scan_out ,nullpin);

    //Input PAD-----
    PDIDGZ IPAD_CLK( .PAD(pad_clk) , .C(clk) );
    PDIDGZ IPAD_RST( .PAD(pad_rst) , .C(rst) );
    PDIDGZ IPAD_SCAN_EN( .PAD(pad_scan_en) , .C(scan_en) );
    PDIDGZ IPAD_SCAN_IN( .PAD(pad_scan_in) , .C(scan_in) );
    PDIDGZ IPAD_NULL_PIN0( .PAD(pad_nullpin[0]) , .C(nullpin[0]) );
    PDIDGZ IPAD_NULL_PIN1( .PAD(pad_nullpin[1]) , .C(nullpin[1]) );
    PDIDGZ IPAD_NULL_PIN2( .PAD(pad_nullpin[2]) , .C(nullpin[2]) );

    //Output PAD-----
    PDT16DGZ OPAD_OUT0( .PAD(pad_out[0]) , .I(out[0]) );
    PDT16DGZ OPAD_OUT1( .PAD(pad_out[1]) , .I(out[1]) );
    PDT16DGZ OPAD_OUT2( .PAD(pad_out[2]) , .I(out[2]) );
    PDT16DGZ OPAD_OUT3( .PAD(pad_out[3]) , .I(out[3]) );
    PDT16DGZ OPAD_SCAN_OUT( .PAD(pad_scan_out) , .I(scan_out) );
endmodule
```

```
module cnt ( out, clk, rst, scan_en, scan_in, scan_out ,nullpin);
    output [3:0] out;
    input clk, rst, scan_en, scan_in;
    input [2:0] nullpin;
    output scan_out;
    wire n33, n34, n35, N2, N3, N4, \add_8/carry[3] , \add_8/carry[2] , n10,
        n15, \out[3] , n17, n28, n31, n37, n38;
    assign scan_out = \out[3] ;
    assign out[3] = \out[3] ;
endmodule
```

# 加入PAD到Gate-Level Verilog Code

在TSMC 0.18um製程下，Input的腳位統一都使用PDIDGZ PAD，Output的腳位統一都使用PDT16DGZ PAD，參數中.PAD()都是CHIP對外的腳位。

```
module cnt_chip ( pad_out, pad_clk, pad_rst, pad_scan_en, pad_scan_in, pad_scan_out ,pad_nullpin);
    output [3:0] pad_out;
    output pad_scan_out;
    input pad_clk, pad_rst, pad_scan_en, pad_scan_in;
    input [2:0] pad_nullpin;

    wire [3:0] out;
    wire clk, rst, scan_en, scan_in;
    wire [2:0] nullpin;
    wire scan_out;
```

Design Vision合出來的Gate-Level Module

```
//Chip Kernel-----
cnt k1( out, clk, rst, scan_en, scan_in, scan_out ,nullpin);
```

```
//Input PAD-----
PDIDGZ IPAD_CLK( .PAD(pad_clk) , .C(clk) );
PDIDGZ IPAD_RST( .PAD(pad_rst) , .C(rst) );
PDIDGZ IPAD_SCAN_EN( .PAD(pad_scan_en) , .C(scan_en) );
PDIDGZ IPAD_SCAN_IN( .PAD(pad_scan_in) , .C(scan_in) );
PDIDGZ IPAD_NULL_PIN0( .PAD(pad_nullpin[0]) , .C(nullpin[0]) );
PDIDGZ IPAD_NULL_PIN1( .PAD(pad_nullpin[1]) , .C(nullpin[1]) );
PDIDGZ IPAD_NULL_PIN2( .PAD(pad_nullpin[2]) , .C(nullpin[2]) );
```

PAD的Instance Name,之後ioc檔的腳位方位都以這個做對映

```
//Output PAD-----
PDT16DGZ OPAD_OUT0( .PAD(pad_out[0]) , .I(out[0]) );
PDT16DGZ OPAD_OUT1( .PAD(pad_out[1]) , .I(out[1]) );
PDT16DGZ OPAD_OUT2( .PAD(pad_out[2]) , .I(out[2]) );
PDT16DGZ OPAD_OUT3( .PAD(pad_out[3]) , .I(out[3]) );
PDT16DGZ OPAD_SCAN_OUT( .PAD(pad_scan_out) , .I(scan_out) );
```

```
endmodule
```

# 加入PAD到Gate-Level Verilog Code

PAD:後面填CHIP Module裡的各個PAD的Instance Name。

建議VSS跟VDD的腳位都放在每邊的中間。

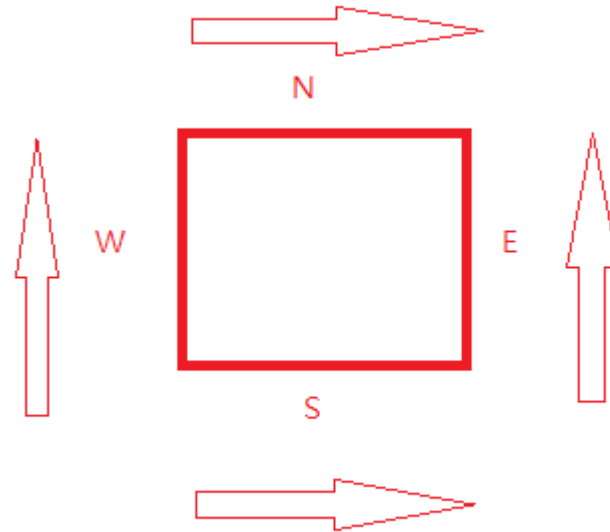
```
1 Version: 1
2
3 Pad: CORNER0      NW      PCORNER
4 Pad: IPAD_CLK      N
5 Pad: PAD_CoreVSS1  N      PVSS1DGZ
6 Pad: PAD_CoreVDD1  N      PVDD1DGZ
7 Pad: IPAD_RST      N
8
9 Pad: CORNER1      NE      PCORNER
10 Pad: OPAD_OUT0     W
11 Pad: OPAD_OUT1     W
12 Pad: OPAD_OUT2     W
13 Pad: OPAD_OUT3     W
14
15 Pad: CORNER2      SW      PCORNER
16 Pad: IPAD_NULL_PIN0 S
17 Pad: PAD_IOVDD1    S      PVDD2DGZ
18 Pad: PAD_IOVSS1    S      PVSS2DGZ
19 Pad: IPAD_NULL_PIN1 S |
20
21 Pad: CORNER3      SE      PCORNER
22 Pad: IPAD_SCAN_EN  E
23 Pad: IPAD_SCAN_IN  E
24 Pad: IPAD_NULL_PIN2 E
25 Pad: OPAD_SCAN_OUT E
```

**VSS VDD** (lines 5-6)  
**IOVSS IOVDD** (lines 17-18)

PAD角落

# 加入PAD到Gate-Level Verilog Code

PAD佈局順序



# 撰寫SDC File

SDC(Synopsys Design Constraints)是一些合成設定的檔案，SOC Encounter可以根據檔案的敘述做一些相關設定。

```
set sdc_version 1.8
current_design cnt_chip 填目前的module name
create_clock [get_ports pad_clk] -name CLK -period 10 -waveform {0 5}
set_clock_uncertainty 0.3 [get_clocks CLK]
set_clock_latency 0.1 [get_clocks CLK]
set_input_delay -clock CLK 1.5 [get_ports pad_rst]
set_drive 1 [get_ports pad_rst]
set_output_delay -clock CLK 1.5 [all_outputs]
set_load 5 [all_outputs]
```

指定clk port  
照DV時所設定的Clock參數  
因為input除了clk只有rst  
所以只指定rst  
dv中設定的輸入輸出負載是以電路與PAD輸入輸出負載  
而我們這邊所設定是以CHIP對外的輸入輸出負載  
單位為k歐姆

# 準備完成

確認CHIP.v、SDC File、IOC File，都準備完成就能進行下一章節。

```
module cnt_chip ( pad_out, pad_clk, pad_rst, pad_scan_en, pad_scan_in, pad_scan_out ,pad_nullpin);
output [3:0] pad_out;
output pad_scan_out;
input pad_clk, pad_rst, pad_scan_en, pad_scan_in;
input [2:0]pad_nullpin;

wire [3:0] out;
wire clk, rst, scan_en, scan_in;
wire [2:0]nullpin;
wire scan_out;

//Chip Kernel-----
cnt k1( out, clk, rst, scan_en, scan_in, scan_out ,nullpin);

//Input PAD-----
PDI16DGZ IPAD_CLK( .PAD(pad_clk) , .C(clk) );
PDI16DGZ IPAD_RST( .PAD(pad_rst) , .C(rst) );
PDI16DGZ IPAD_SCAN_EN( .PAD(pad_scan_en) , .C(scan_en) );
PDI16DGZ IPAD_SCAN_IN( .PAD(pad_scan_in) , .C(scan_in) );
PDI16DGZ IPAD_NULL_PIN0( .PAD(pad_nullpin[0]) , .C(nullpin[0]) );
PDI16DGZ IPAD_NULL_PIN1( .PAD(pad_nullpin[1]) , .C(nullpin[1]) );
PDI16DGZ IPAD_NULL_PIN2( .PAD(pad_nullpin[2]) , .C(nullpin[2]) );

//Output PAD-----
PDT16DGZ OPAD_OUT0( .PAD(pad_out[0]) , .I(out[0]) );
PDT16DGZ OPAD_OUT1( .PAD(pad_out[1]) , .I(out[1]) );
PDT16DGZ OPAD_OUT2( .PAD(pad_out[2]) , .I(out[2]) );
PDT16DGZ OPAD_OUT3( .PAD(pad_out[3]) , .I(out[3]) );
PDT16DGZ OPAD_SCAN_OUT( .PAD(pad_scan_out) , .I(scan_out) );]

endmodule

module cnt ( out, clk, rst, scan_en, scan_in, scan_out ,nullpin);
output [3:0] out;
input clk, rst, scan_en, scan_in;
input [2:0]nullpin;
output scan_out;
wire n33, n34, n35, N2, N3, N4, \add_8/carry[3] , \add_8/carry[2] , n10,
n15, \out[3] , n17, n28, n31, n37, n38;
assign scan_out = \out[3] ;
assign out[3] = \out[3] ;

SFFFSRX2 out_reg_3 ( .D(N4) , .SI(n37) , .SE(scan_en) , .CK(clk) , .SN(1'b1) ,
.RN(n10) , .Q(n28) );
SFFFSRX2 out_reg_0 ( .D(n31) , .SI(scan_in) , .SE(scan_en) , .CK(clk) , .SN(
1'b1) , .RN(n10) , .Q(n35) , .QN(n31) );
SFFFSRX2 out_reg_2 ( .D(N3) , .SI(n38) , .SE(scan_en) , .CK(clk) , .SN(1'b1) ,
.RN(n10) , .Q(n33) , .QN(n37) );
SFFFSRX2 out_reg_1 ( .D(N2) , .SI(n31) , .SE(scan_en) , .CK(clk) , .SN(1'b1) ,
.RN(n10) , .Q(n34) , .QN(n38) );
CLKINX4 U12 ( .A(rst) , .Y(n10) );
BUFX12 U13 ( .A(n34) , .Y(out[1]) );
BUFX12 U14 ( .A(n33) , .Y(out[2]) );
ADDX4 U15 ( .A(out[2]) , .B(\add_8/carry[2] ) , .C0(\add_8/carry[3] ) , .S(N3)
```

```
set sdc_version 1.8

current_design cnt_chip

create_clock [get_ports pad_clk] -name CLK -period 10 -waveform {0 5}

set_clock_uncertainty 0.3 [get_clocks CLK]

set_clock_latency 0.1 [get_clocks CLK]

set_input_delay -clock CLK 1.5 [get_ports pad_rst]

set_drive 1 [get_ports pad_rst]

set_output_delay -clock CLK 1.5 [all_outputs]

set_load 5 [all_outputs]
```

Version: 1

Pad: CORNER0	NW	PCORNER
Pad: IPAD_CLK	N	
Pad: IPAD_RST	N	
Pad: PAD_CoreVSS1	N	PVSS1DGZ
Pad: PAD_CoreVDD1	N	PVDD1DGZ

Pad: CORNER1	NE	PCORNER
Pad: OPAD_OUT0	W	
Pad: OPAD_OUT1	W	
Pad: OPAD_OUT2	W	
Pad: OPAD_OUT3	W	

Pad: CORNER2	SW	PCORNER
Pad: IPAD_NULL_PIN0	S	
Pad: IPAD_NULL_PIN1	S	
Pad: PAD_IOVDD1	S	PVDD2DGZ
Pad: PAD_IOVSS1	S	PVSS2DGZ

Pad: CORNER3	SE	PCORNER
Pad: IPAD_SCAN_EN	E	
Pad: IPAD_SCAN_IN	E	
Pad: IPAD_NULL_PIN2	E	
Pad: OPAD_SCAN_OUT	E	