Impact Of Reset Domain Crossing On PowerPro Tool

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(VLSI Design)

By

Ritika Modi

(15MECV20)

तमसो मा ज्योतिर्गमय

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED ‘A’ GRADE

Electronics & Communication Engineering Department
Institute of Technology
NIRMA University
Ahmedabad-382 481
May 2017

Impact Of Reset Domain Crossing On PowerPro Tool

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Ritika Modi

(15MECV20)

Under the guidance of

External Project Guide:	**Internal Project Guide:**
Mrs. Divya Parihar	**Dr. N.M.Devashrayee**
PowerPro PV Team Manager	Institute of Technology
Mentor, A Siemens Business	Nirma University
Noida	Ahmedabad

Electronics & Communication Engineering Department
Institute of Technology
NIRMA University
Ahmedabad-382 481
May 2017

तमसो मा ज्योतिर्गमय

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED ‘A’ GRADE

Certificate

This is to certify that the Major Project entitled **“Impact Of Reset Domain Crossing On PowerPro tool ”** submitted by **Ritika Modi (15MECV20)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design , NIRMA University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge,haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. N. M. Devashrayee
Internal Guide

Dr. N. M. Devashrayee
PG Coordinator (VLSI Design)

Dr D K Kothari
Head, EC Dept.

Dr. Alka Mahajan
Director, IT-NU

Date:

Place: Ahmedabad

Mentor®
A Siemens Business

Certificate

This is to certify that the Project entitled "**Impact Of Reset Domain Crossing On PowerPro Tool**" submitted by **Ritika Modi (15MECV20)**, towards the submission of the Project for requirements for the degree of Master of Technology in VLSI Design, NIRMA University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mrs. Divya Parihar
Manager-PowerPro PV,
CSD Calypto-PowerPro
Mentor, A Siemens Business
Noida

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at NIRMA University and has not been submitted elsewhere for a degree.

b. Due acknowledgement has been made in the text to all other material used

-Ritika Modi

Acknowledgement

It gives me immense pleasure to acknowledge **Mr. Abhishek Ranjan**, Engineering Director, and **Mrs. Divya Parihar**,PowerPro PV Team Manager at Mentor,A Siemens Business, For providing me a platform to explore my abilities I would have never succeeded in my thesis without the cooperation, encouragement and help provided to me by various people.

I would like to express my sincere gratitude to **Dr. Alka Mahajan** (Director of Institute of technology Nirma university , Ahmedabad) and **DR. D.K.Kothari** (Head of Department of Electronics communication Engineering Department NIRMA University, Ahmedabad) for his continuous guidance and support. I Would like to take this opportunity to thank **Dr. N. M. Devashrayee** (Professor and Program Coordinator, M. Tech - EC (VLSI Design)),and my Internal Guide and all the faculties for their vision, support, and encouragement to provide me with the opportunity to carry out my project work in such a renowned and esteemed organization.

I am deeply indebted to my thesis supervisor,Mrs. Neha Babel and my other team members : MR. Jitender Mor, Ms. Kriti kumari,Mrs. Sakshi Choudhary , Mr. Anmol Garg and Mr. Hitesh Dhingra , For there constant motivation regarding the project and also for thert constant guidance, supervision, kind co-operation, and invaluable support in all aspects.Their wisdom, clarity of thought and support motivated me to bring this project to its present state.

- Ritika Modi
15MECV20

Abstract

Power consumption is one of the top concerns of Very Large Scale Integration (VLSI)circuit design, for which Complementary Metal Oxide Semiconductor (CMOS) is the primary technology. Today's focus on low power is not only because of the recent growing demands of mobile applications. Even before the mobile era, power consumption has been a fundamental problem. To solve the power dissipation problem,many researchers have proposed different ideas from the device level to the architectural level and above. However, there is no universal way to avoid tradeoffs between power, delay and area, and thus designers are required to choose appropriate techniques that satisfy application and product needs.

In general, low power VLSI Design can be achieved at all levels of the VLSI Design (system, algorithm, architecture, circuit, logic, device, technology levels). But optimizations for low power VLSI Design done at higher abstraction results in comparatively higher power savings. The tool used for power optimization also effects some other parameters, if not taken care may lead to metastability.

During Power Optimization technique, some factors needs to be taken care, like metastability , so that expected result does not vary. Metastability effects the output of design , Thus some factors like clock domain crossing & reset domain crossing should be considered. synchronizers when used removes the metastability due to clock domain crossing . Similarly Reset Domain crossing also leads to metastability when source flop is asserted with reset while destination flop does not have reset assertion at the same time. Those scenarios needs to be worked on , where reset domain crossing condition is added by PowerPro tool in clock gating or reset hierarchies.At the end, flow was automated to run with multiple designs so as to error out when differences in orig RTL and power optimized RTL was observed .Flow for Reset Domain Crossing violation, analysis, challenges & results observed are mentioned in this report.

Contents

List of Figures

Chapter 1

PowerPro And Questa Tool

1.1 PowerPro tool

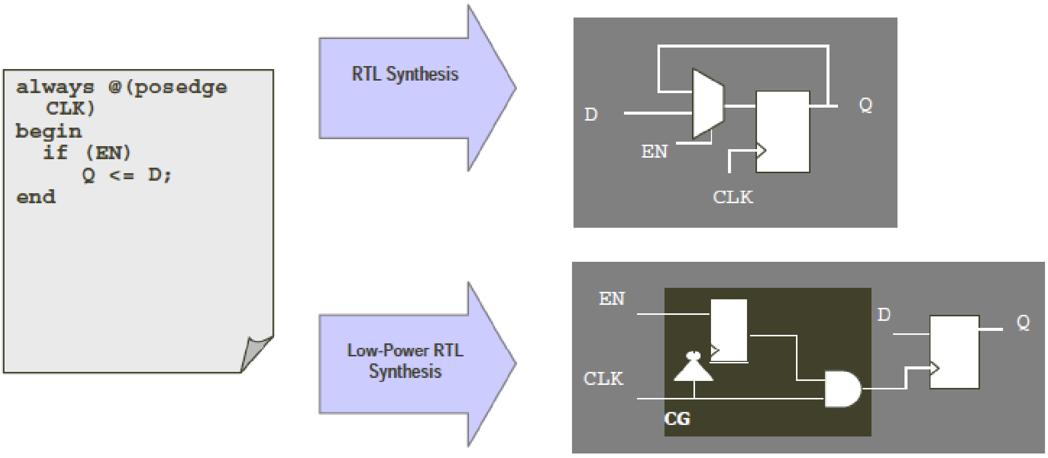
1.1.1 Introduction

PowerPro tool outputs the power-optimized RTL. For reduction in total power, dynamic & static power needs to be reduced.Dynamic power is a function of transistors switching power.By optimizing at every stage of design to the extent it is possible, goal for power optimization is achieved. Optimization for power done at a higher level has a greater impact. One of the popular technique for dynamic power optimization is clock gating.

1.1.2 Combinational Clock Gating

By identifying that condition when data held in a register is not used further down the line, clock is shut off for that period thus, lead to combinational clock gating. In figure, new data is enabled in register Q when EN signal is High. Combinational clock gating is substituted when tool identifies this pattern.

In figure new data is enabled in register Q when EN signal is High . Combinational clock gating is substituted when tool identifies this pattern .

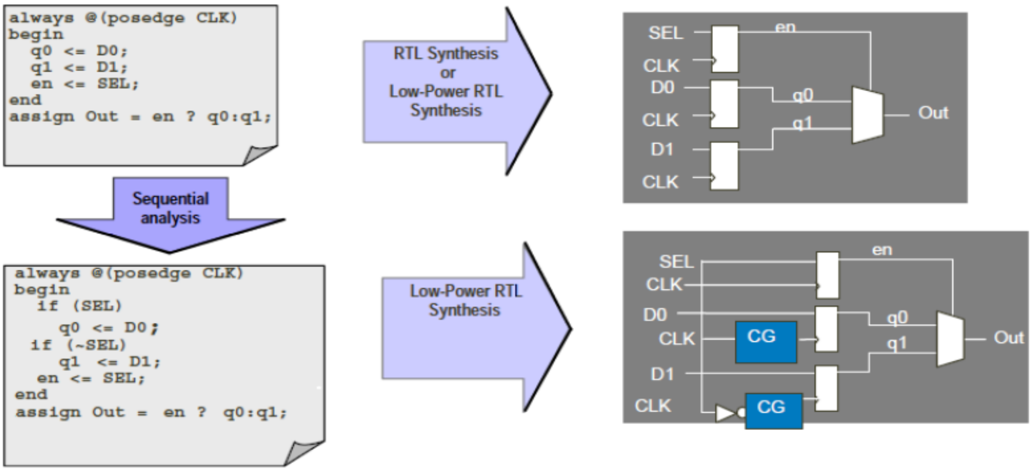


```
always @(posedge
   CLK)
begin
  if (EN)
     Q <= D;
end
```

Figure 1.1: Combinational Clock Gating

1.1.3 Sequential Clock Gating

Newly enabled condition is generated to hold the data if new data is not required down the flow when data is constant or invalid.

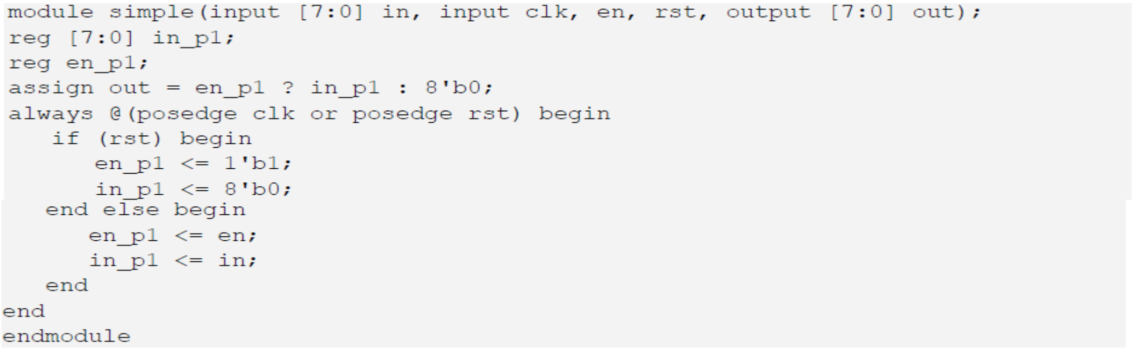


```
always @(posedge CLK)
begin
  q0 <= D0;
  q1 <= D1;
  en <= SEL;
end
assign Out = en ? q0:q1;
```

```
always @(posedge CLK)
begin
  if (SEL)
    q0 <= D0;
  if (~SEL)
    q1 <= D1;
  en <= SEL;
end
assign Out = en ? q0:q1;
```

Figure 1.2: Sequential clock gating

In the figure, we notice that register q0 & q1 always latch a new data with a cycle. We noticed that when EN is high, data latched to q1 is not used further down the line thus, by connecting en logic of q1 to IJsel of mux data is not latched further thus reduces its power. This technique is more global in nature as it offers high power savings. It is referred as STB-c (constant) or STB-s(symbolic).



```
module simple(input [7:0] in, input clk, en, rst, output [7:0] out);
reg [7:0] in_p1;
reg en_p1;
assign out = en_p1 ? in_p1 : 8'b0;
always @(posedge clk or posedge rst) begin
   if (rst) begin
      en_p1 <= 1'b1;
      in_p1 <= 8'b0;
   end else begin
      en_p1 <= en;
      in_p1 <= in;
   end
end
endmodule
```

Figure 1.3: Original RTL code

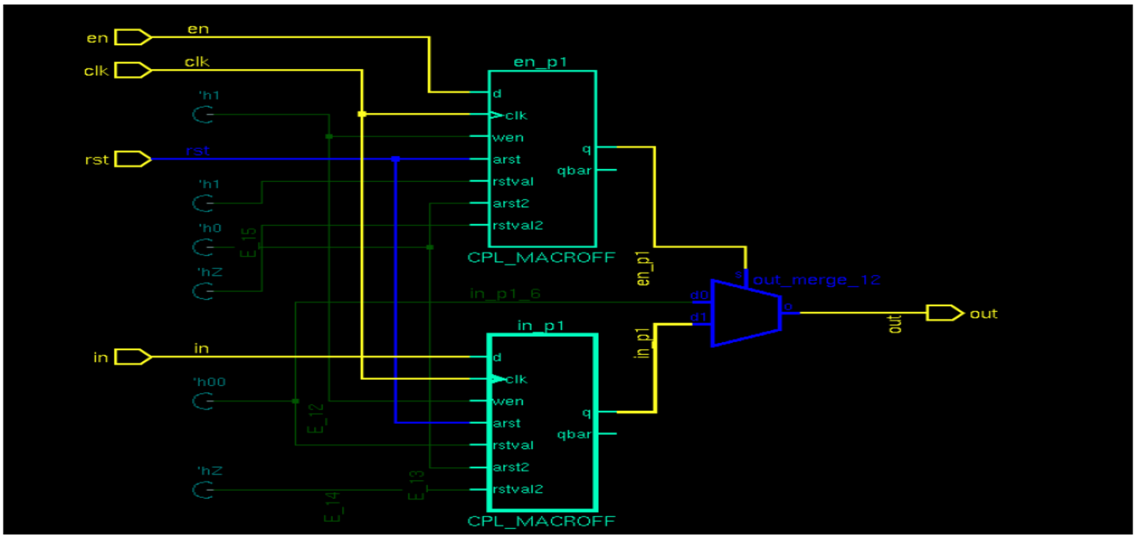
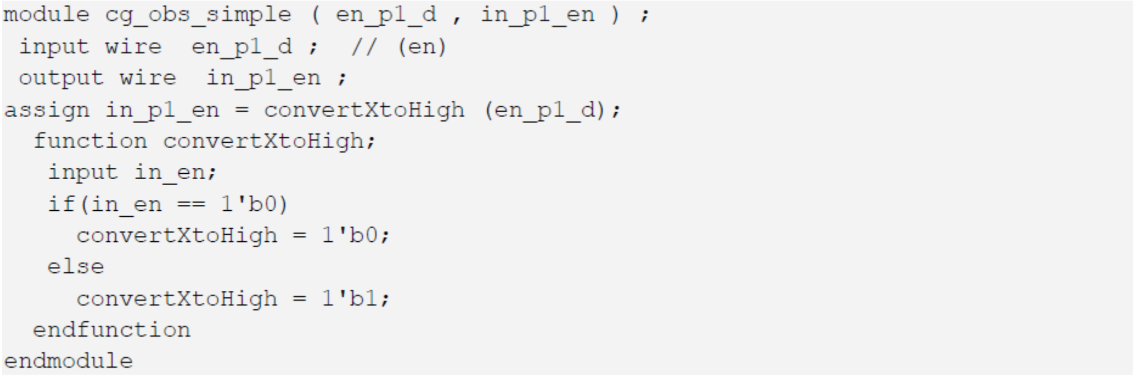


Figure 1.4: Schematic of above code without clock gating

When clock gating condition is matched , new enabled logic is added to the RTL to have optimized Power RTL as output. Similarly, for memories , when same data is repeatedly read without any intervening write, all data except the first one can be gated off , it is referred as memory stability gating.



```
module cg_obs_simple ( en_p1_d , in_p1_en ) ;
 input wire  en_p1_d ;  // (en)
 output wire  in_p1_en ;
assign in_p1_en = convertXtoHigh (en_p1_d);
  function convertXtoHigh;
   input in_en;
   if(in_en == 1'b0)
     convertXtoHigh = 1'b0;
   else
     convertXtoHigh = 1'b1;
  endfunction
endmodule
```

Figure 1.5: New enabled logic

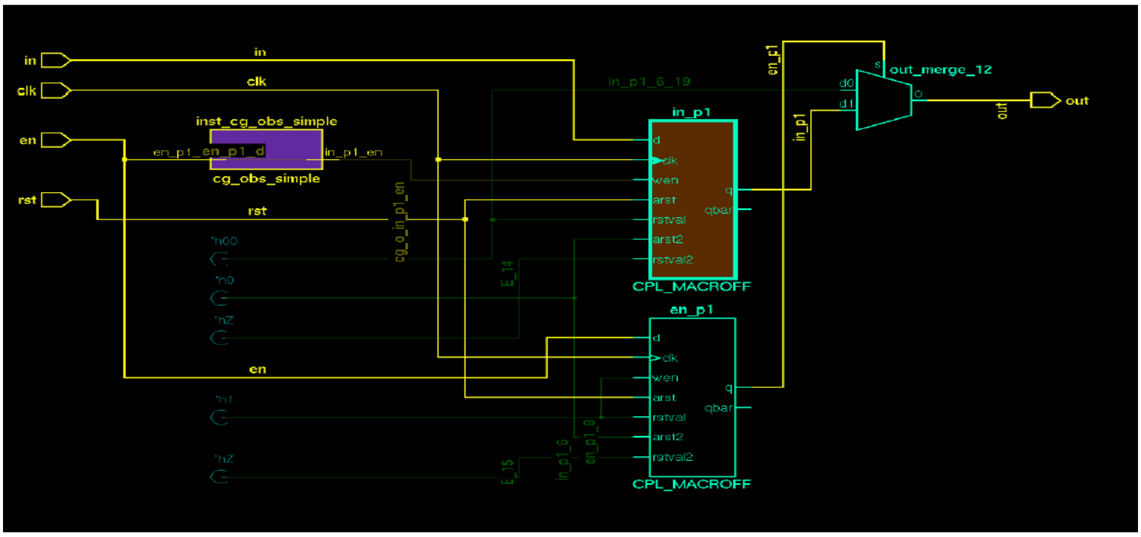


Figure 1.6: Power optimized RTL

1.1.4 PowerPro Clock Gating Design Flow

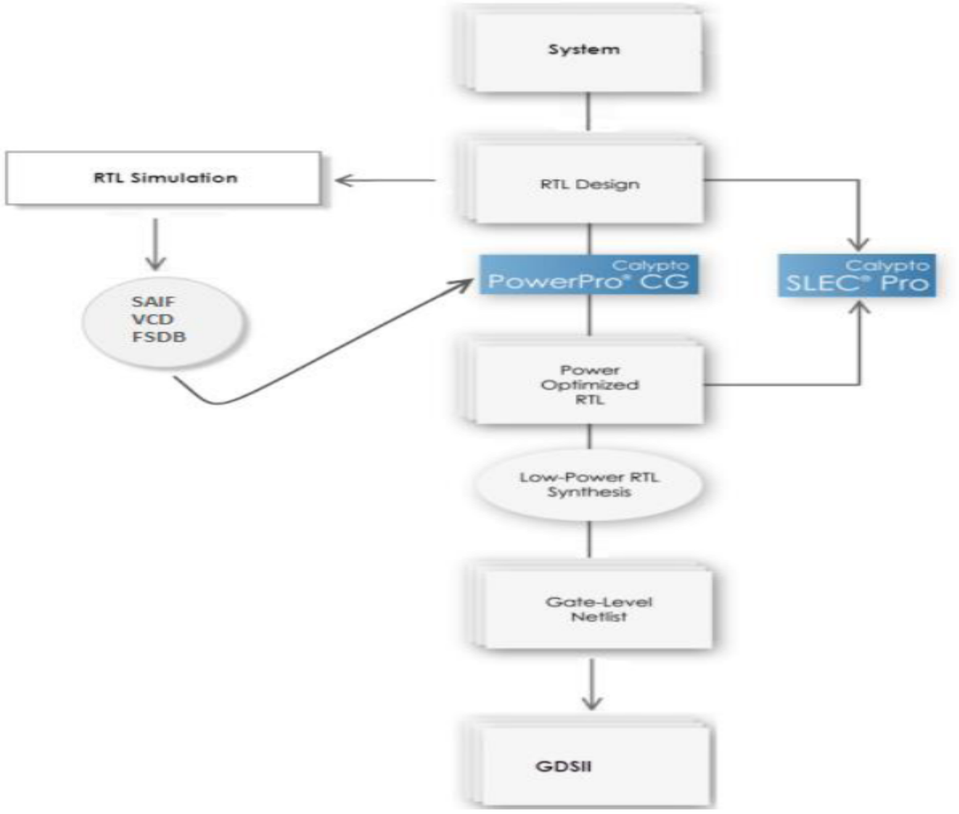


Figure 1.7: PowerPro in design flow stages

PowerPro tool inserts enable logic to optimize clock gate insertion before logic synthesis. The figure shows the typical flow. PowerPro requirements are same as logic synthesis tools, optionally it might need library Files, switching activity information and other constraints like clock information, reset information etc .

1.2 Questa Tool

1.2.1 Questa Flow

Questa tool flow is mentioned below with its flow & description.

Vlib:To set accessibility for existing libraries & to create a design library for use by vlog/vmap/vcom.

vmap: logical-to-physical mapping of a design library, also to remove or report current mapping.

verror: Reports common front-end utility error

vcom: To compile VHDL source file to design or resource library.

vlog: Compile verilog & system verilog to design or resource library.

qverify: To run questa functional verification flow. Qverific is used here to check reset domain crossing violations.

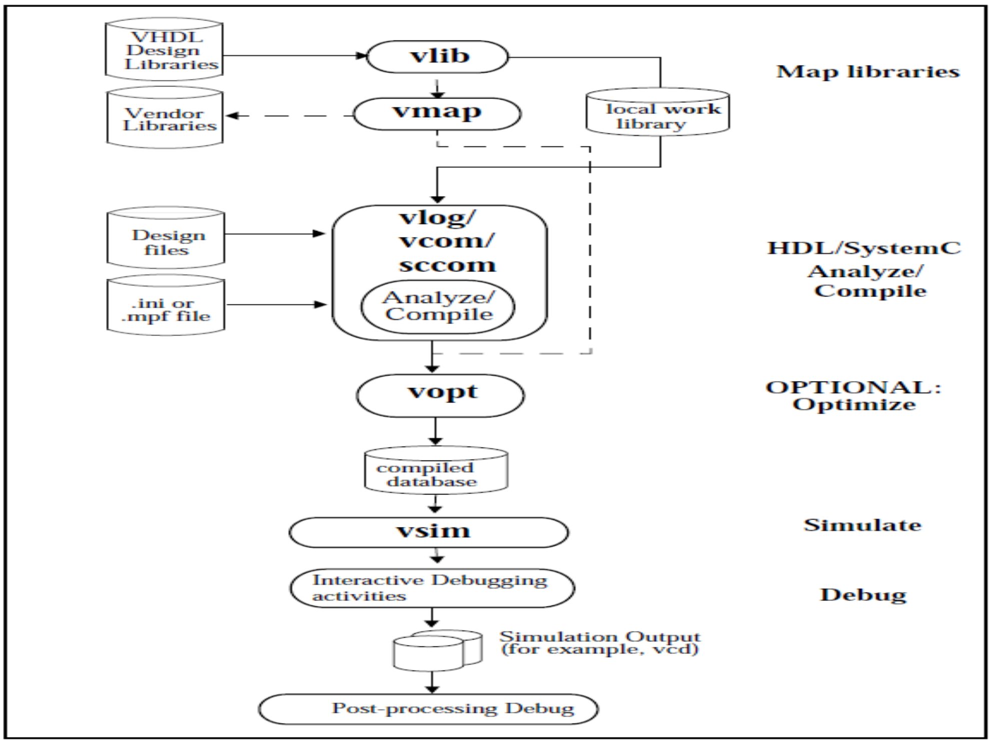


Figure 1.8: Questa Flow

Questa reset check is used to check RDC violation, verifies reset signalling network, taking RTL as input & performs an exhaustive bottom-up reset tree analysis inferring all the reset structure of a design.

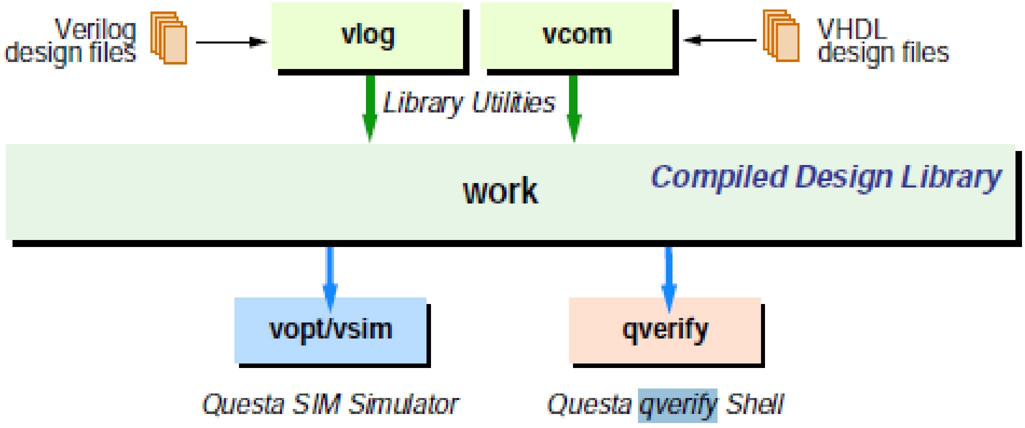


Figure 1.9: Compiler Commands

Qverify requires a "do" file with the list of commands, where reset check is also added to analyse the reset domain crossing violations. Also, a directive file with information about clocks its period resets type active_high or active_low is also passed to "do" file.

Chapter 2

Effect Of Reset Domain Crossing On Power Optimized RTL

2.1 Introduction

Decreasing device sizes and an increase in complex designs lead to multimillion transistor system running with multiple asynchronous resets. These SOC systems have multiple interfaces. Several modern interfaces are inherently asynchronous from rest of the chip.Sometimes there is also a need for designing major sub-blocks of SOCs to run on independent resets. The cross-reset domain crossing signal poses a unique and challenging issue for verification.

Other conditions when such resets are needed include when the software requests a warm reset to the system and the contents of the System RAM are expected to retain their values, as well as when there is a reset event causing a global system reset

2.2 Metastability

The proper operation of a clocked flip-flop depends on the input being stable for a certain period of time before and after the clock edge. If the setup and hold-time violation requirements are met, the correct output will appear at a valid output level. However if the setup time and hold time violations are not met, the output of the flip-flop may take much longer time to reach a valid logic level. This unstable behaviour is called metastability.

As described in the figure, here metastability occurred because flops are in different clock domain. It also occurs when flops are in different reset-domain.

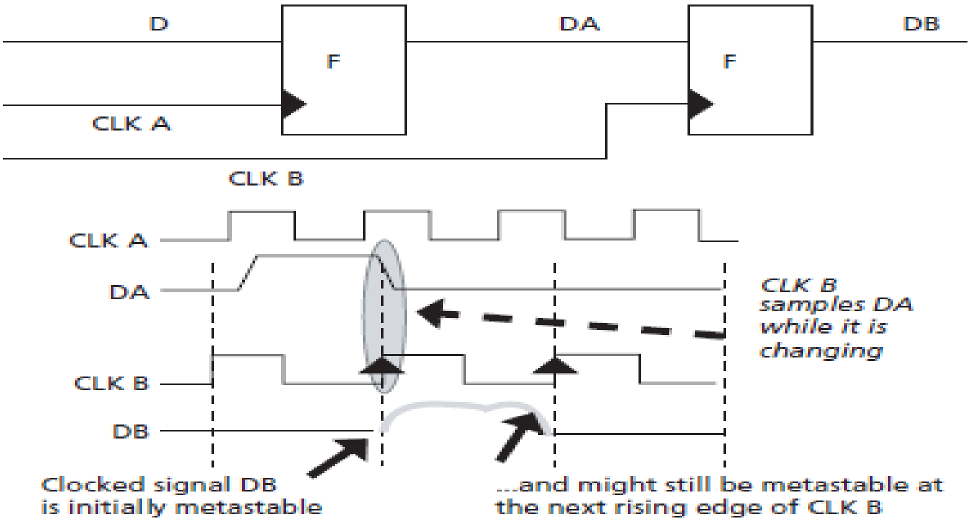

D
F
DA
F
DB
CLK A
CLK B
CLK A
DA
CLK B
DB
CLK B samples DA while it is changing
Clocked signal DB is initially metastable
...and might still be metastable at the next rising edge of CLK B

Figure 2.1: Clock-Domain-Crossing

2.3 Synchronizer

A synchronizer samples asynchronous signals & outputs a signal that is synchronized to transition of local or sampled clock. The simplest & most common synchronizer used is two flip-flop synchronizer

A synchronizer samples asynchronous signals & outputs a signal that is synchronised to the transition of local or sampled clock. The simplest & most common synchronizer used is the two-flip-flop synchronizer The asynchronous input signal sampled by first flop is passed to the new clock domain & waits for the complete cycle to check metastability on stage1 output signal to decay, this signal is sampled by the same

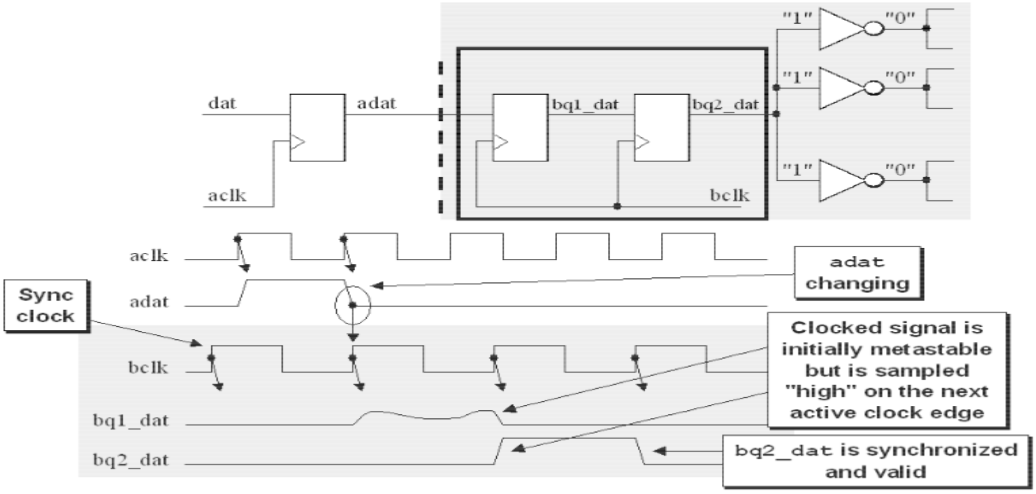

dat
adat
aclk
bq1_dat
bq2_dat
bclk
"1"
"0"
aclk
adat
bclk
bq1_dat
bq2_dat
Sync clock
adat changing
Clocked signal is initially metastable but is sampled "high" on the next active clock edge
bq2_dat is synchronized and valid

Figure 2.2: Two flop synchronizer

clock to the second stage flop, with the goal that 2nd stage is now stable. For most scenarios two-flop synchronizer technique is sufficient to remove metastability. The probability for time between synchronisation failure (MTBF) is dependent on many variables including frequency of clocks.
synchronizers are solutions to reduce the metastability due to clock domain crossing. Like clock Domain crossing, reset domain crossing also lead to metastability which is explained below .

2.4 Need For Multiple Resets

There are reasons why multiple resets are necessary for any system-on-chip design. The most important are that in any such design there is always some functionality which needs to be active up to the time the device gets a cold reset (Power On Reset due to unavailability of the Power supply). These include such things as timekeeping functionality, calendaring features, clock and reset control modules. They are all expected to be intact when a global reset (warm reset) is asserted.

2.5 Introduction To Reset Domain Crossing

When only first flip-flop undergoes reset & thus injecting metastability at destination flop as asynchronous change of first flop output lead to Reset Domain Crossing condition.

2.5.1 RDC Structure Identification In Design

In figure both flops are in different reset domain. when reset async_rstA_b is asserted while other reset async_rstA_b connected to destination flop is not asserted then output of destination flop may be metastable, thus if it's output is used further down the line, may lead to functional error.

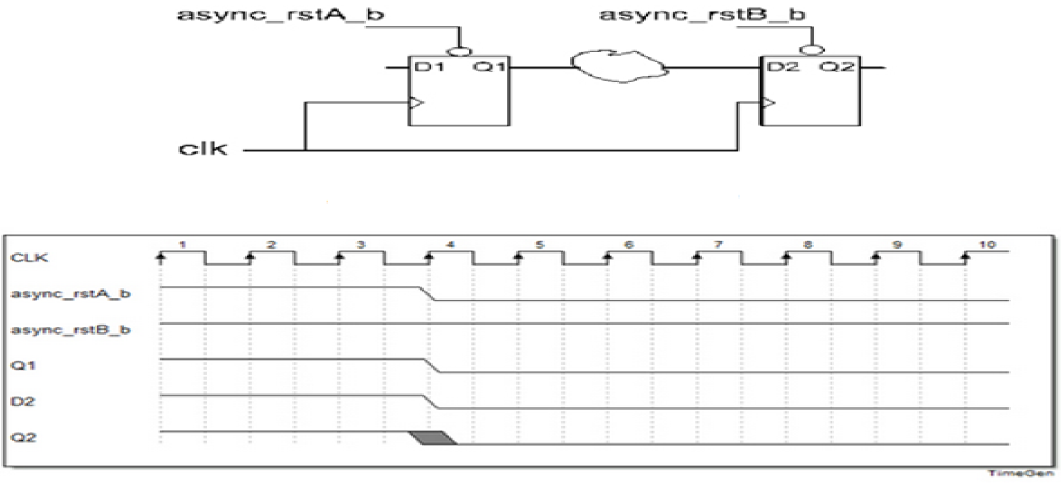


Figure 2.3: Reset-Domain-Crossing

2.5.2 Source Flop - Physically Superset

In this scenario when source flop reset is physically superset of destination flop reset as, two resets "AND'ed" , have its output connected to flop D1, the source flop thus, assertion on source flop may occur sooner then the destination flops, which may again lead to metastability & error at functionality level.

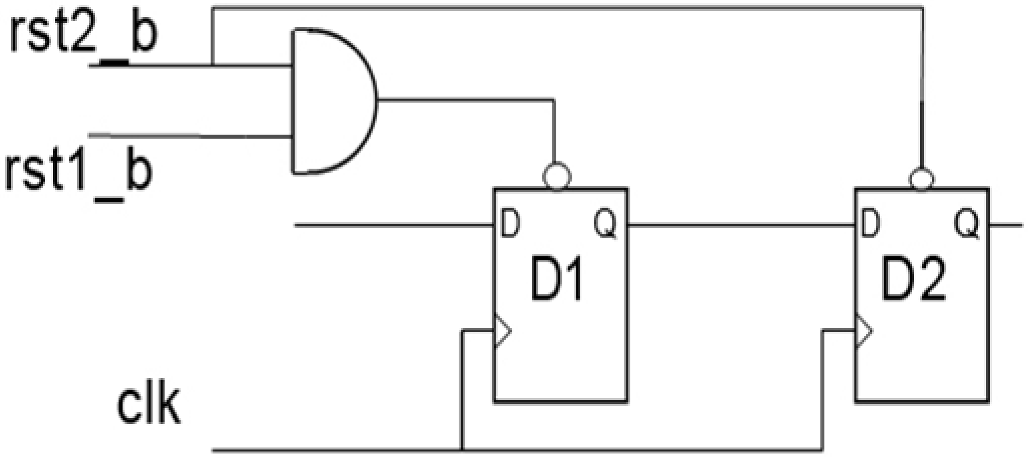


Figure 2.4: Reset domain crossing (physically superset reset)

2.5.3 Metastability At Module Configuration

Functionality that should be reset only when POR occur and should function during other resets (warm or global resets) . Hence , registers for these functionality should remain intact during other resets .However, if global/warm reset occur while write operation , content may get corrupted due to metastability.

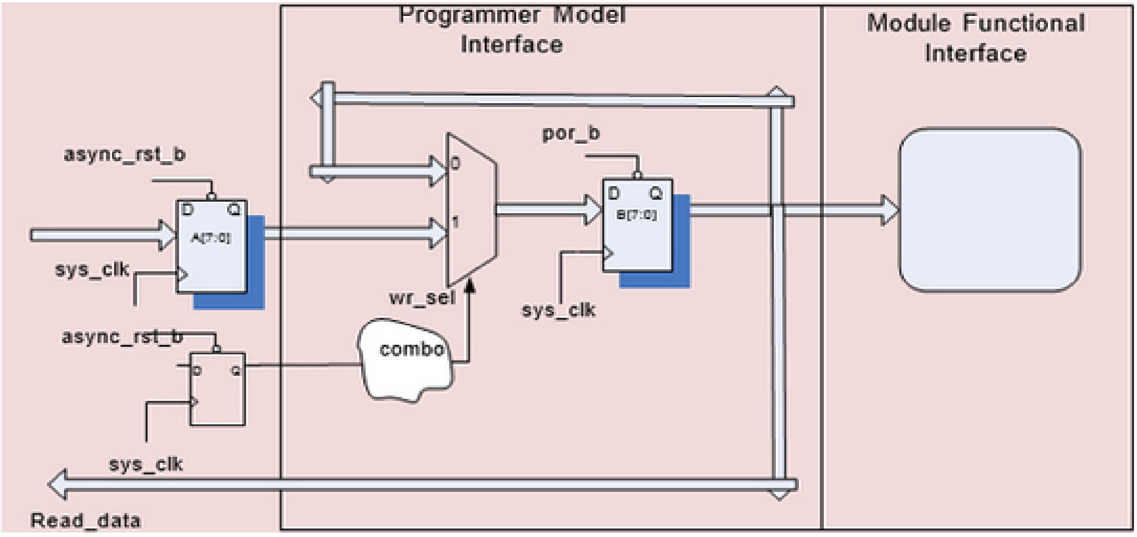


Figure 2.5: Reset domain crossing (RDC) issue while programming configuration register

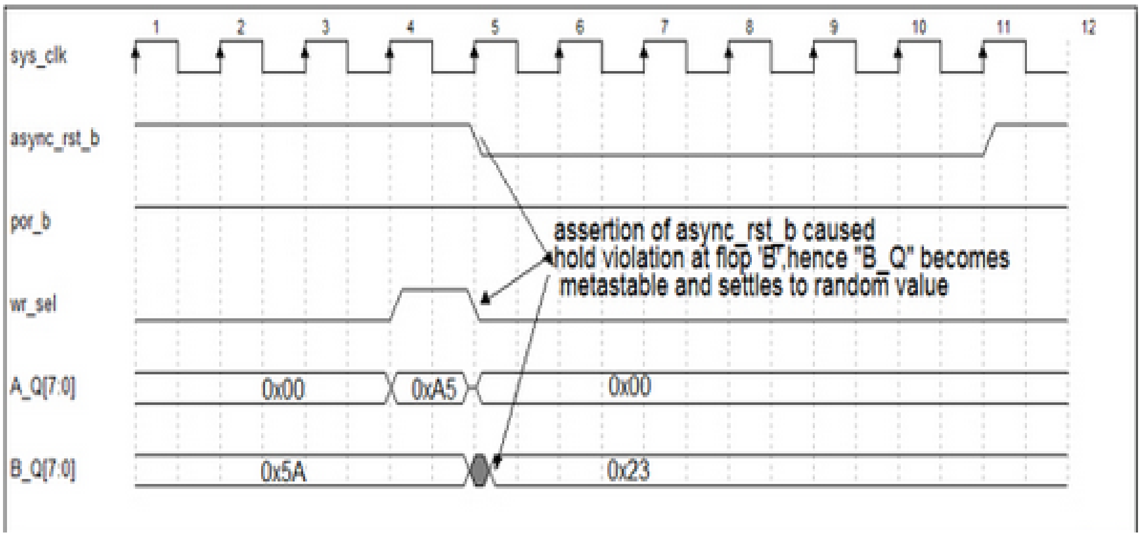


Figure 2.6: Waveform depicting above issue

Chapter 3

Flow & EDA Tools Used for Experiment

3.1 Introduction

The basic idea behind the experiment is to check the Reset domain crossing effect effect on PowerPro tool . The original RTL & patched optimized design through PowerPro tools results are checked for Reset domain crossing violation, through questa flow . The increase in the violation in power optimized design as compared to original RTL shows the new addition of Reset Domain Crossing condition in the design added by tool for power optimizations.

The RDC violation check was done through Questa tool. Questa reset check verifies reset signaling network , taking RTL as input & performs an exhaustive bottom-up reset tree analysis inferring all the reset structure of a design.

Results include reports on the synchronicity , polarity & set/reset functionality of all reset tree register nodes & RDC registers within the same clock domain & the complete matrix of clock & reset signals .

3.2 Methodology Adopted

Analysis of some unit testcases helped to understand the scenarios where occurrence of reset domain crossing violations was observed.These Unit testcases with original RTL as input was passed through questa flow to check the reset domain crossing vioKltions . This design was also passed as input with some more constraints to powerPro tool that outputs the power optimized RTL . The output of power optimized RTL is further given as input to questa flow to reproduce the violation list for optimized RTL.

The process mentioned above , helped understand that there are scenarios where questa reports reset domain crossing violations for the flops in clock gating hierarchies added by PowerPro for power optimizations .

Further the same flow as mentioned above was practiced for big customer designs & to observe if there are cases where reset domain crossing violation is increased from original to power optimized RTL.

After experimenting with many customer's designs, increase in reset domain crossing violation from original to patched design was reported for some cases.

To handle the reset domain crossing violation due to flops patched by PowerPro in clock gating or PowerPro reset hierarchies a global was set to kill the moves on flops when fanins of these flops are from flops within hierarchies added by tools for power optimization.

Complete flow was repeated with global enable to check it's effect initially on unit testcases . For detailed analysis Customer's designs were rerun with global set and also without enabling the global.

The QOR difference between the two methods for all designs were observed & customer designs were selected from them , so that further analysis could be done for these cases .

After going through the above mentioned method some scenarios where observed moves were not killed even with global enabled. All these cases were reproduced on unit testcases for detailed analysis.

At last , complete flow was automated to run the script with multiple designs so as to report all the designs with increased violations in optimized design as compared to original RTL.

3.3 RDC Flow Automation

Flow was automated with two methods ,

1.Script is executed after all designs were run in regressions , script uses the result of it further to check RDC violation , thus works serially . 2.Script is run in parallel with each single design that is passed PowerPro in regression .

First Method :

This script is executed at the same path where regression results are stored .Initially , after questa is run with vlog & vmap , list of path for rtl_mod_[verilog/vhdl].f is stored in a flie . *.f File path present is passed to vlog one by one , after scratch path in the *.f file is replaced . Top module is grepped after questa vmap is run & is passed as input to qverific 'do' file .A violation_list file at the end of the execution of script will have a list of violations with their design name and log file path .

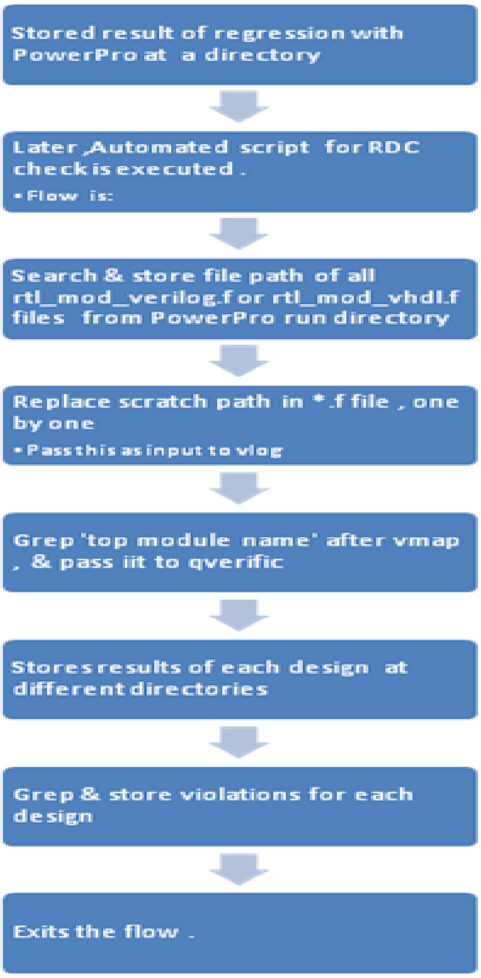

Stored result of regression with PowerPro at a directory
Later ,Automated script for RDC check is executed .
• Flow is:
Search & store file path of all rtl_mod_verilog.f or rtl_mod_vhdl.f files from PowerPro run directory
Replace scratch path in *.f file , one by one
• Pass this as input to vlog
Grep 'top module name' after vmap , & pass iit to qverific
Stores results of each design at different directories
Grep & store violations for each design
Exits the flow .

Figure 3.1: Automated script flow method 1

Disadvantages are :

1.It requires more time & space also, flow is hard coded .

Second method :

In this flow, automated script is sourced with the design , with required constraints like library needed, clock information etc, needed as input to PowerPro tool .

The sourced script hacks the read design flow, passes it's arguments to the proc where design is read in similar method, the way it was read before hacking the session .

Later, a file with questa flow information related to vlib, vmap vlog & qverific is dumped, where original RTL read in PowerPro is passed as input to vlog. Questa qverific 'do' file requires the top module, which is also passed from the design that was read before & output directory for a qverific run is set as rdc_result_orig.

Qverific also requires the directive that should have information about clocks & resets, mentioned with the detail commands like for all clock , clock name with time period. Similarly, for resets, name of reset & its type, active high/low needs to be present.

To write the information about clocks name & period also resets name & type ,have grepped from the Powerpro run log file , from the variables that store this information . This directive file is also passed to qverific both time when original & patched designs are passed as input to vlog .

After rdc script file was dumped, clock gating is enabled to have power optimized RTL as output. Depending on the moves, whether stability - s or C or both are asserted with newly enabled logic, 'rtl_mod_verilog.f' The file path is searched & passed to the new proc which appends the initially written RDC script with vlog & qverific options, where vlog has the new power optimized RTL as input & qverific result is stored at directory rdc_result_patched.

Till now, questa RDC flow is only dumped in a script . After clock gating insertion is completed, rdc script will be executed & its run will be stored in another file so that one can check detail information of questa run flow for both designs.

After execution of RDC script, checks on violations are added to assert error if the difference in violation occur, else will display message in log file showing number of violations in design & exits the flow.

This script was sourced with designs run in regressions of a suit where we need to report the difference in violations of original & power optimized design.

The output of this script when run in regression of a suit is compared with the hard coded script to verify the results.

In next chapter Experiment, results & observations are mentioned that describes the methodology mentioned here.

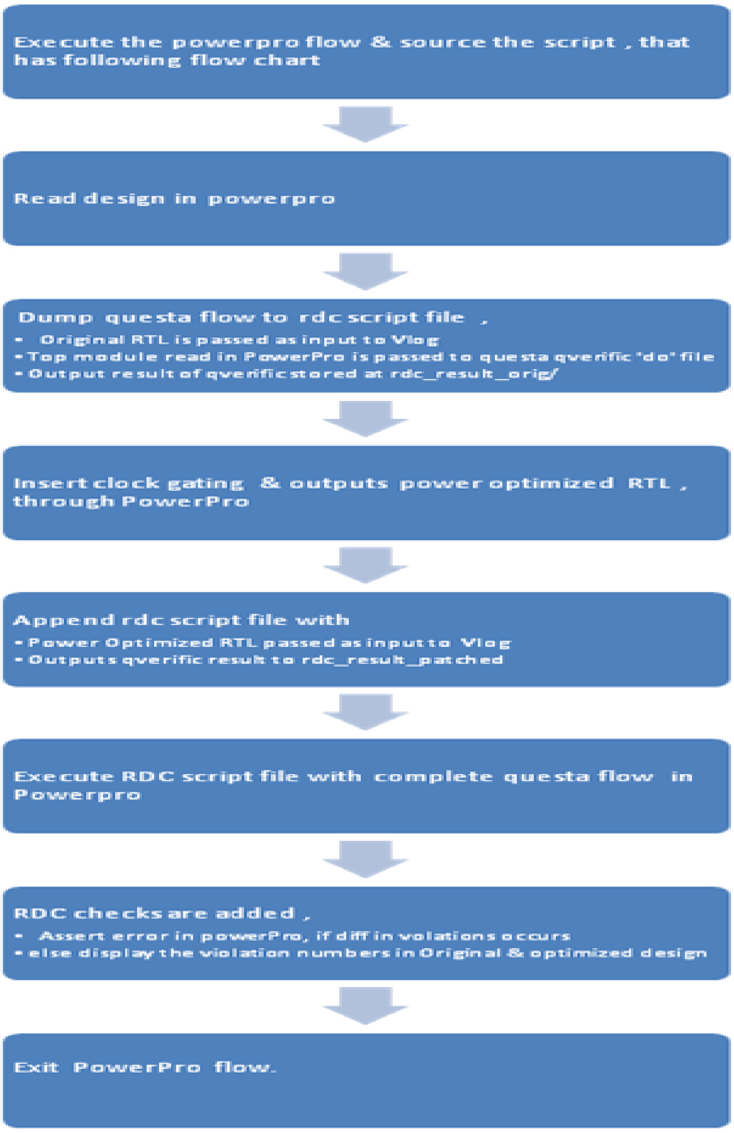

Execute the powerpro flow & source the script , that has following flow chart
Read design in powerpro
Dump questa flow to rdc script file ,
• Original RTL is passed as input to Vlog
• Top module read in PowerPro is passed to questa qverific "do" file
• Output result of qverific stored at rdc_result_orig/
Insert clock gating & outputs power optimized RTL , through PowerPro
Append rdc script file with
• Power Optimized RTL passed as input to Vlog
• Outputs qverific result to rdc_result_patched
Execute RDC script file with complete questa flow in Powerpro
RDC checks are added ,
• Assert error in powerPro, if diff in violations occurs
• else display the violation numbers in Original & optimized design
Exit PowerPro flow.

Figure 3.2: Automated script flow for method 2

Chapter 4

Experiment Results & Observations

4.1 Experiment Results

Initially, to understand the scenarios ,if reset domain crossing violation occurs on the flops of clock gating heirarchies or reset hierarchies added by tool, many unit testcases were created further analysis was done to summarise the occurrance of those cases. also, to understand those scenarios where no further increase in violations was observed . In this chapter two cases are described in detail with the complete flow followed, also analysis of these cases with the global enabled , which ideally should have solved this issue. Results with the difference in output by setting/unsetting the global final observations for these two cases are mentioned. ALso, results for the experiments done on one of the customer's design is also described at the end .

4.2 Observations

4.2.1 Unit Testcase Scenario1

Scenario 1: Five flops with alternate same async resets

Original RTL

```
module simple(in1,clk,enable,reset_a,reset_b,reset_c,out);
input [15:0]in1;
input clk,reset_a,reset_b,reset_c,enable;
output reg [15:0] out;
reg [15 :0] in_p1;
reg [15 :0] in_p2;
reg [15 :0] in_p3;
reg [15 :0] in_p4;
always@(posedge clk or posedge rese_a) begin
if(reset_a)begin
in_p1<=16’b0;
end
else
begin
if (enable)
begin
in_p1<=in1;
end
end
end
always@(posedge clk or posedge reset_b) begin
if(reset_b)begin
in_p2<=16’b0;
end
else
begin
begin
in_p2<=in_p1;
end
```

```
end
end
always@(posedge clk or posedge reset_a) begin
if(reset_a)begin
in_p3<=16’b0;
end
else
begin
begin
in_p3<=in_p2;
end
end
end
always@(posedge clk or posedge reset_b) begin
if(reset_b)begin
in_p4<=16’b0;
end
else
begin
begin
in_p4<=in_p3;
end
end
end
always@(posedge clk or posedge reset_c) begin
if(reset_c)begin
out<=16’b0;
end
else
```

```
begin
begin
out<=in_p4;
end
end
end
endmodule
```

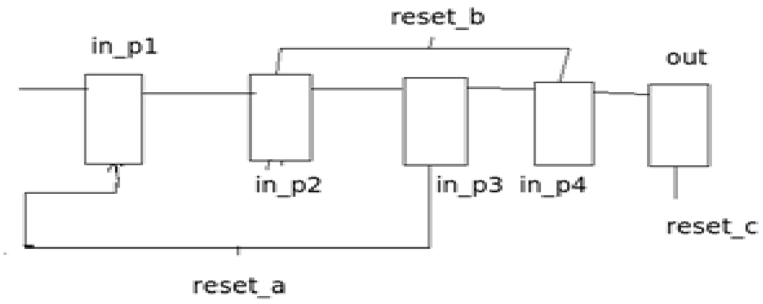


Figure 4.1: Orig RTL for case1

All flops are in same clock domain: clk
Input: in
Flops in_p1 & in_p3 are in reset domain reset_a
Flops in_p2 & in_p4 are in reset domain reset_b
Flop out are in reset domain reset_c.

Patched result

In this type of scenario, we figure out that violations with respect to original RTL has increased by 1. Original RTL has 4 Violation, while power optimized RTL has 5 violation .Clock gating & reset hierarchies are marked with purple, these are added

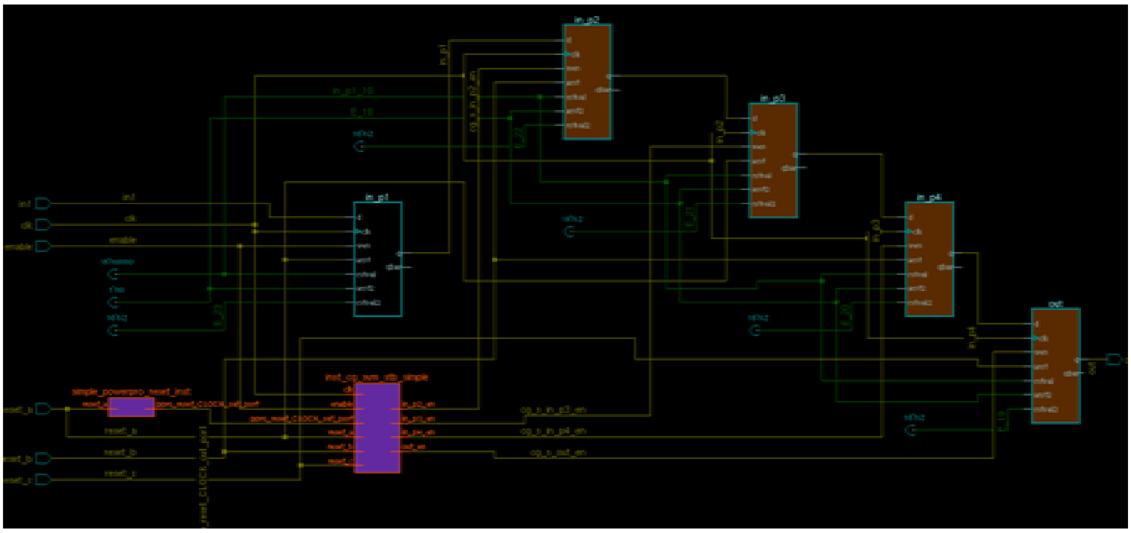


Figure 4.2: Power optimized RTL for case1

by PowerPro for power optimization.

Flops of instance Inst_cg_sym_stb_simple Sprop_0_in_p4 & inst_cg_sym_stb_simple

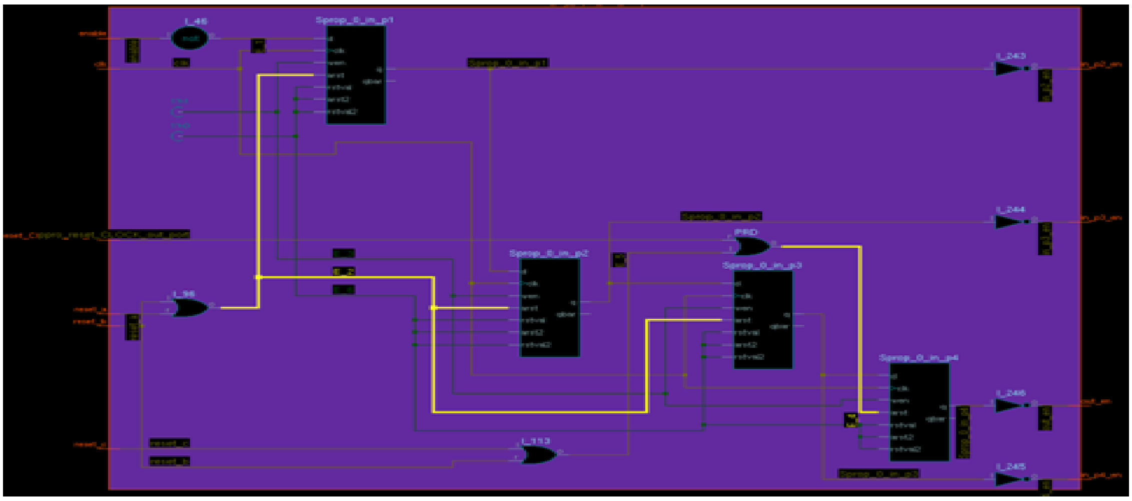


Figure 4.3: Patched RTL for case1

Sprop_0_in_p3 are connected through different resets though output of one flop is connected to other through data path, thus it will lead to metastability and is marked as RDC violation flops.

Questa result RDC violation list for above scenario is :

reset_a<A,H>: clk: start: in_p1

reset_b<A,H>: clk: end: in_p2

reset_b<A,H>: clk: start: in_p2

reset_a<A,H>: clk: end: in_p3

reset_a<A,H>: clk: start: in_p3

reset_b<A,H>: clk: end: in_p4

reset_b<A,H>: clk: start: in_p4

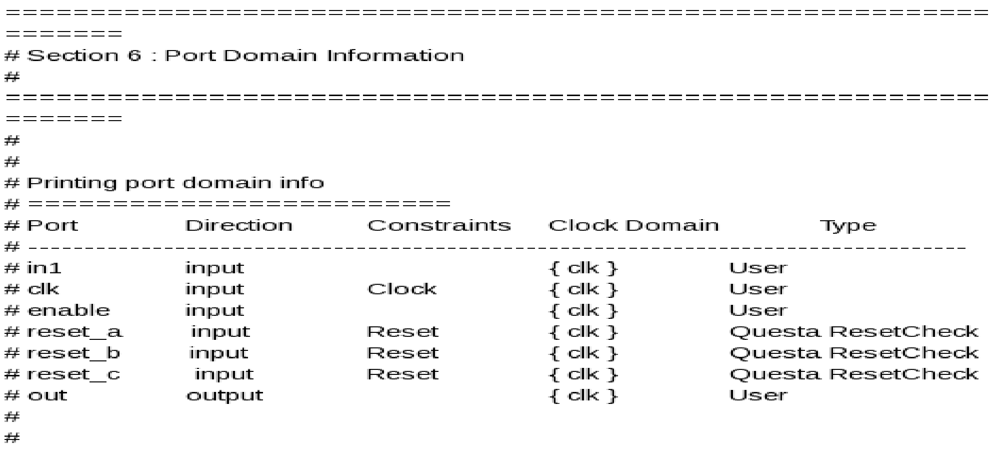
reset_c<A,H>: clk: end: out

inst_cg_sym_stb_simple.E_2<A,H>: clk: start :

inst_cg_sym_stb_simple.Sprop_0_in_p3

inst_cg_sym_stb_simple.E_4<A,H>: clk: end:

inst_cg_sym_stb_simple.Sprop_0_in_p4



```
========================================================
=======
# Section 6 : Port Domain Information
#
========================================================
=======
#
#
# Printing port domain info
# =========================
# Port          Direction      Constraints    Clock Domain        Type
# ---------------------------------------------------------------------------------------
# in1           input                         { clk }         User
# clk           input          Clock          { clk }         User
# enable        input                         { clk }         User
# reset_a        input         Reset          { clk }         Questa ResetCheck
# reset_b        input         Reset          { clk }         Questa ResetCheck
# reset_c         input        Reset          { clk }         Questa ResetCheck
# out           output                        { clk }         User
#
#
```

Figure 4.4: Port information for case1 from questa

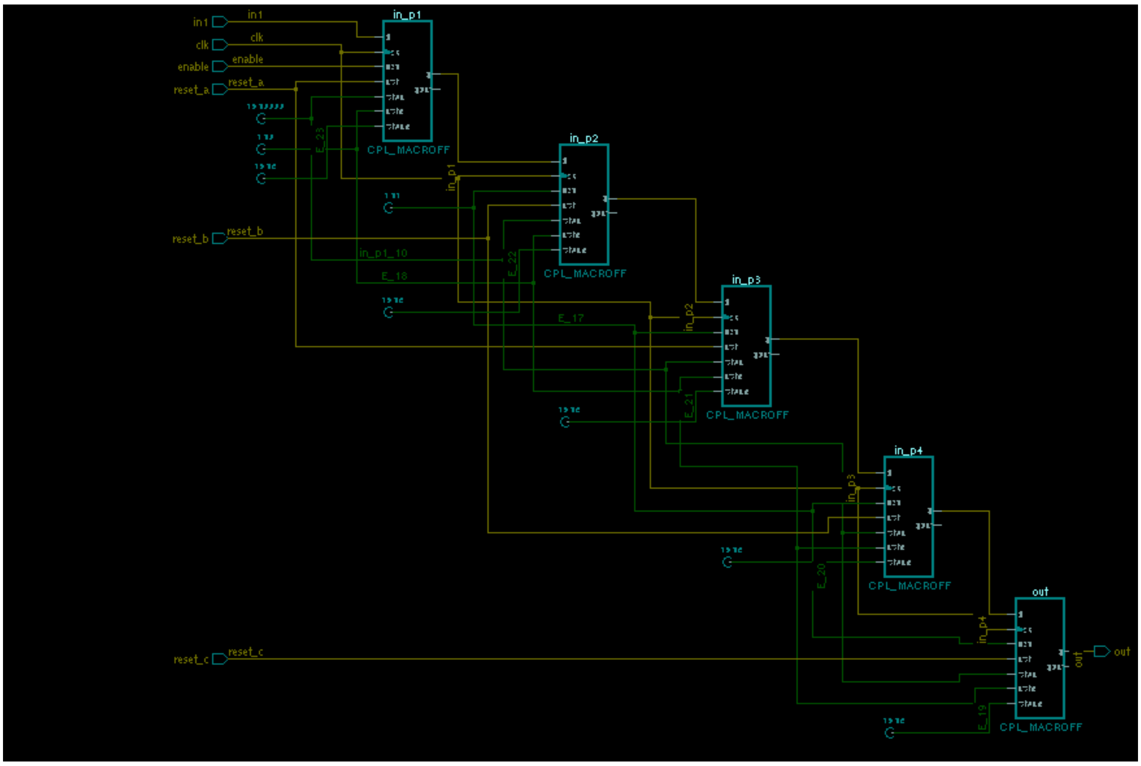


Figure 4.5: Power optimized RTL with global enabled

Result For Scenario1 Enabled With Global:

When same design was enabled with global on, all flops that have moves were committed with flops of clock gating (cg) hierarchy with RDC violations thus, all moves on flops were decomitted. In next case even by enabling global flop's moves won't be decommitted. Complete Procedure for second case is explained below.

4.2.2 Unit Testcase for Scenario2

scenario 2: Three flops with two async , one synchronized reset

original RTL

```
module simple(reset_a,reset_b,clk,select,in,enable,out);
input clk,reset_a,reset_b,select,enable;
input [15:0]in;
output reg [15:0]out;
reg res;
reg [15:0]in_p1;
reg [15:0]in_p2;  always@(posedge clk) begin
if(select)
in_p1<=in;
else
in_p1<=16'b1;
end
always@(posedge clk or posedge reset_a) begin
if(reset_a)
begin
in_p2<=16'b0;
end
else
begin
if(enable)
in_p2<=in_p1;
end
end
always@(posedge clk or posedge reset_b) begin
if(reset_b)
```

```
out<=16'b0;
else
out<=in_p2;
end
endmodule
```

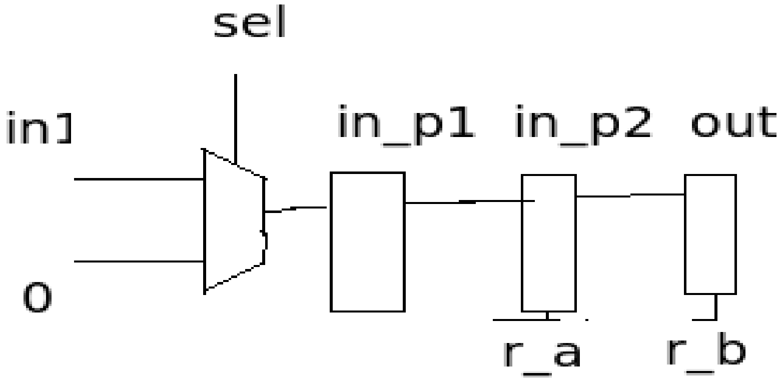


Figure 4.6: Orig RTL for case2

All flops are in clock domain clk

Input in1

In_p1 flop has synchronized reset sel

In_p2 flop is in reset domain r_a

Out flop is in reset domain r_b

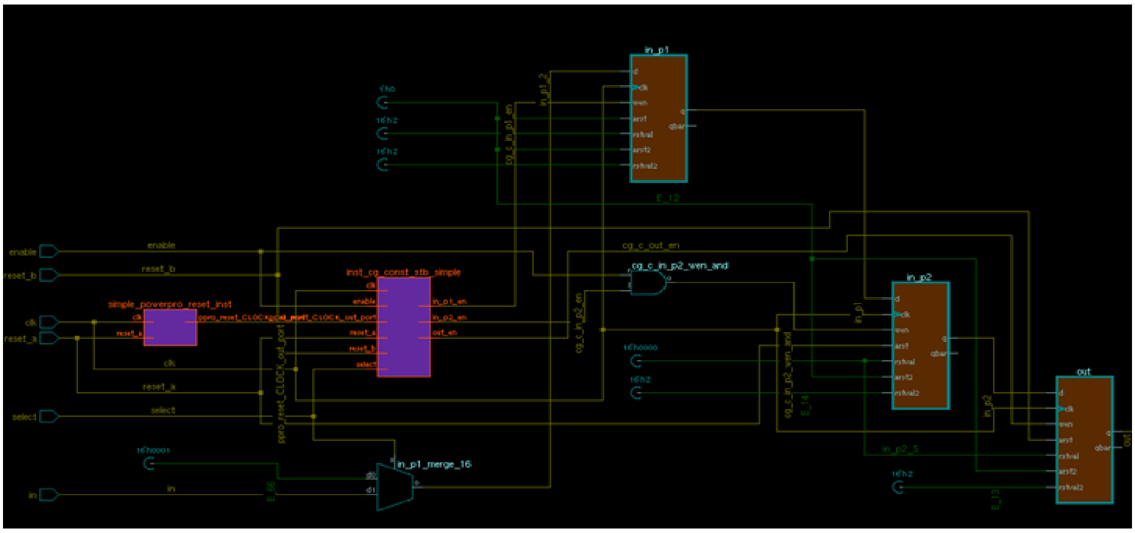


Figure 4.7: Power optimized RTL for case2

In this type of scenario, we figure out that violations with respect to original RTL has increased by 2. Original RTL has 1 Violation, while power optimized RTL has 3 violation.

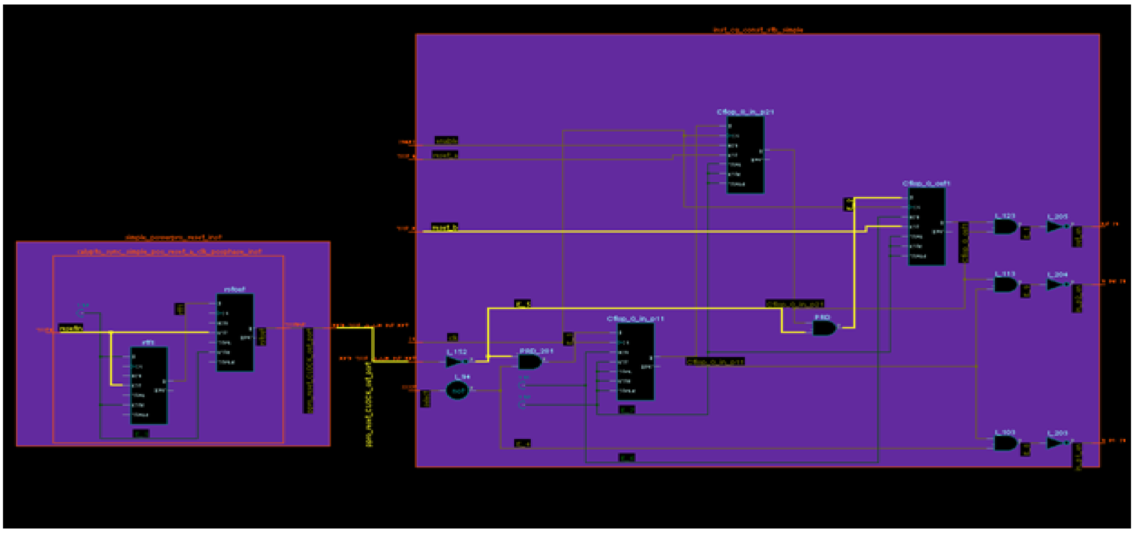


Figure 4.8: Patched RTL for case2

1.Cflop_0_in_p21 & Cflop_0_out1 flops in instance inst_cg_const_stb_simple added by PowerPro are source & destination flops connected with different reset domain reset_a & reset_b, thus these flops are marked during RDC violation.

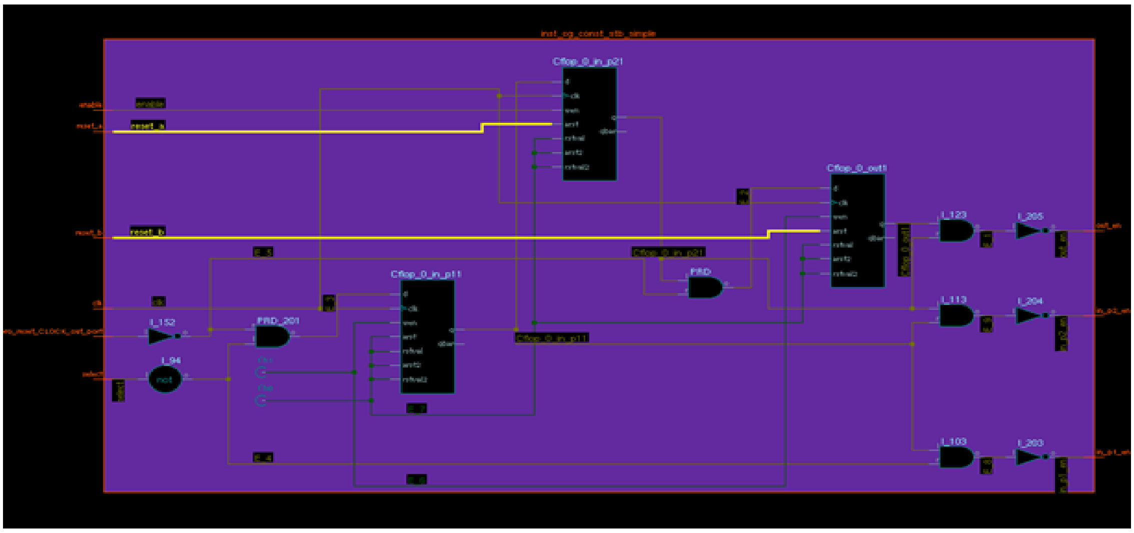
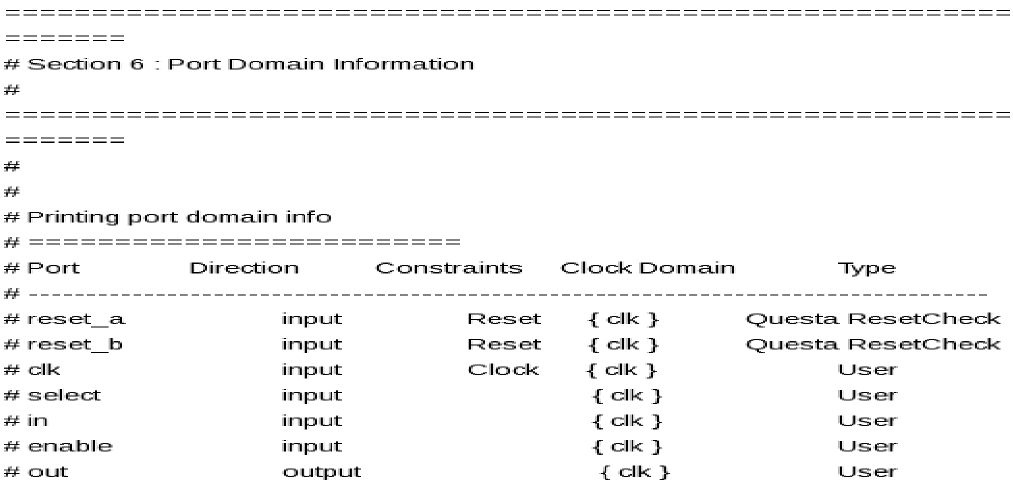


Figure 4.9: Patched RTL for case2

2 Flop rst_out in instance simple_PowerPro_reset_inst and Cflop_0_out1 flops in

instance inst_cg_const_stb_simple are source and destination flops (i.e. result of source flop is connected through datapath to a destination flop) are in different reset domain with reset reset_a & reset_b respectively, thus RDC occur and both these flops are counted during violation.
Above case was one of the scenarios that was observed initially to understand the effect of reset Domain Crossing on power optimized design.



```
=============================================================
=======
# Section 6 : Port Domain Information
#
=============================================================
=======
#
#
# Printing port domain info
# ========================
# Port           Direction        Constraints     Clock Domain          Type
# ---------------------------------------------------------------------------------
# reset_a               input           Reset      { clk }        Questa ResetCheck
# reset_b               input           Reset      { clk }        Questa ResetCheck
# clk                   input           Clock      { clk }              User
# select                input                       { clk }             User
# in                    input                       { clk }             User
# enable                input                       { clk }             User
# out                   output                       { clk }            User
```

Figure 4.10: Port information for case2 from questa

Questa result RDC violation list for above scenario is :

reset_a<A,H> : clk : start : in_p2

reset_b<A,H> : clk : end : out

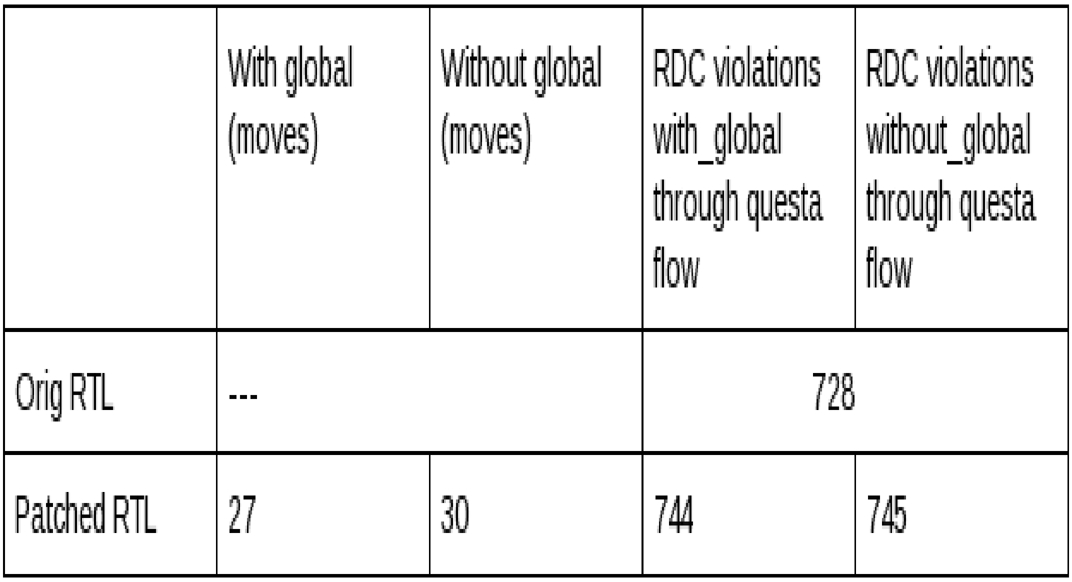
reset_a<A,H>: clk: start: **inst_cg_const_stb_simple.Cflop_0_in_p21**

reset_b<A,H>: clk: end: **inst_cg_const_stb_simple.Cflop_0_out1**

reset_a<Se,A,H>: clk: start:

simple_powerpro_reset_inst.calypto_sync_simple_pos_reset_a-_clk_posphase_inst.rstout

reset_b<A,H>: clk: end: **inst_cg_const_stb_simple.Cflop_0_out1**



	With global (moves)	Without global (moves)	RDC violations with_global through questa flow	RDC violations without_global through questa flow
Orig RTL	---		728	
Patched RTL	27	30	744	745

Figure 4.11: Report difference after seting/unseting global, from PowerPro tool

Analysis of efficiency report of both cases when global was enabled and disabled, to understand the effect of both method on the PowerPro tool.
Now, when same design was enabled with global that should decommitt the moves if clock gating (cg) hierarchy has flops with RDC violations, results are same as above, moves were not killed in this scenario.

Result For Scenario2 with Global On:

To understand and analyse the scenarios where global's effect was not observed, the QOR difference between designs where global was set & unset was observed on all customers designs. Those cases where QOR difference was observed were further analysed & and these cases were further reproduced through unit testcases, so that a detailed analysis could be done.

Observation Of scenario2 On One Of The Customer's Design is:

PowerPro result with global enabled:

STB-s

1.optimizing enable logic for sequential gating domains: 158

2.Clk Gating summary: moves accepted: 27

3.Number of clock-gated flip-flops during power estimation: 11411 (87.65%)

4.Register Power Savings during Symbolic Stability-based Sequential Optimization: 4.11074%

5.Total flops added for this stage: 37

6.New Enable Logic Area: 1704 (0.19 %)

7.New Enable Logic (justified to user signals) Area: 3661 (0.4 %)

8. Optimizing RTL-style flop definitions in the source: 276

Powerpro result with global disabled :

STB-s

1. optimizing enable logic for sequential gating domains: 162

2. Clk Gating summary: moves accepted: 30

3.Number of clock-gated flip-flops during power estimation: 11450 (87.91%)

4.Register Power Savings during Symbolic Stability-based Sequential Optimization: 4.7625%

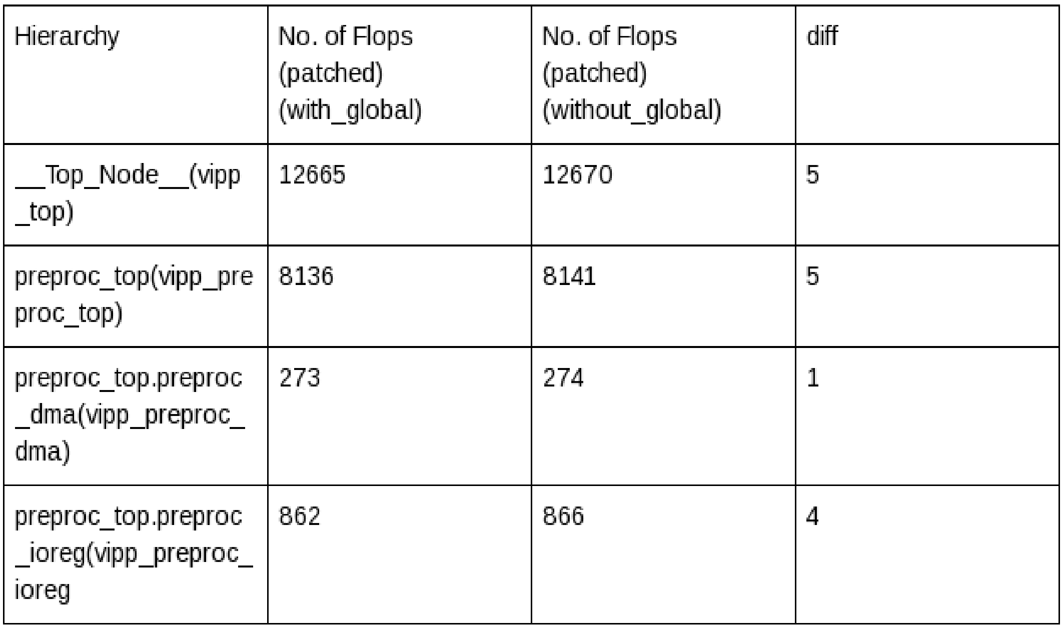
5.Total flops added for this stage: 42

6.New Enable Logic Area: 1900 (0.21 %)

7.New Enable Logic (justified to user signals) Area: 3867 (0.42 %)

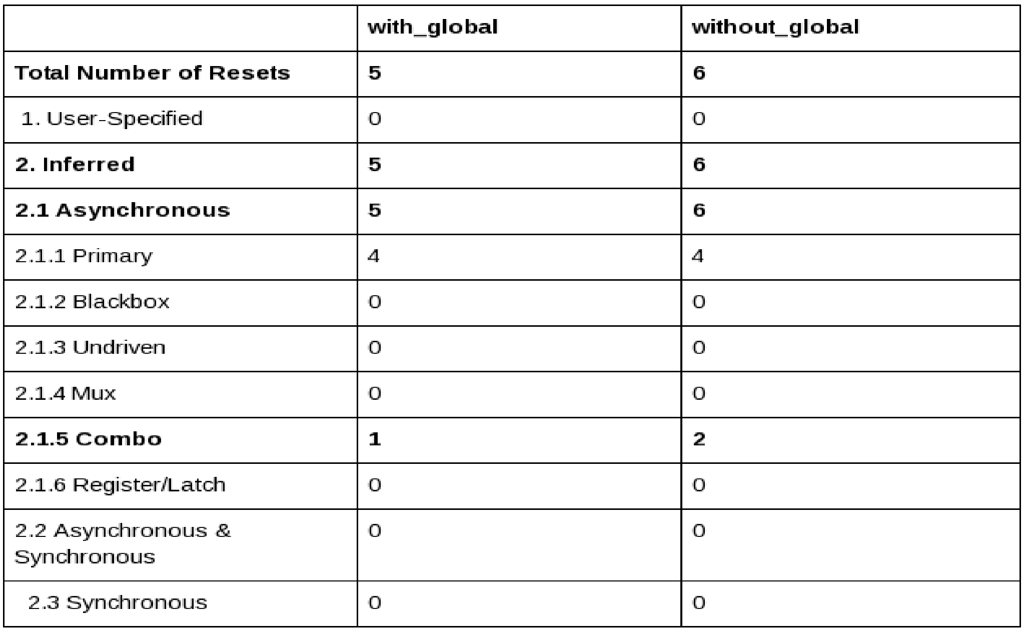
8. Optimizing RTL-style flop definitions in the source: 279

At last, automated flow was run to check result on all customer's design. detail of automated flow was already mentioned in previous chapter.



Hierarchy	No. of Flops (patched) (with_global)	No. of Flops (patched) (without_global)	diff
__Top_Node__(vipp_top)	12665	12670	5
preproc_top(vipp_preproc_top)	8136	8141	5
preproc_top.preproc_dma(vipp_preproc_dma)	273	274	1
preproc_top.preproc_ioreg(vipp_preproc_ioreg	862	866	4

Figure 4.12: Efficiency difference in report by enabling/disabling global



	with_global	**without_global**
Total Number of Resets	**5**	**6**
1. User-Specified	0	0
2. Inferred	**5**	**6**
2.1 Asynchronous	**5**	**6**
2.1.1 Primary	4	4
2.1.2 Blackbox	0	0
2.1.3 Undriven	0	0
2.1.4 Mux	0	0
2.1.5 Combo	**1**	**2**
2.1.6 Register/Latch	0	0
2.2 Asynchronous & Synchronous	0	0
2.3 Synchronous	0	0

Figure 4.13: Difference in reset tree summary report by enabling/disabling global in questa flow

Chapter 5

Conclusion

5.1 Conclusion

Initially, experiments performed over unit designs to analyse those scenarios whether there are cases where RDC Violations increases for the power optimized RTL as compared to original RTL.

Results & observations helped to conclude & understand those cases that lead to increase in the RDC violation after optimization of RTL. With multiple reset domains in designs there were scenarios where flops in instances added by tool occur in different reset domain and thus increase in RDC violation.

Above, mentioned experiment when repeated for the customer's design, to check if these cases also occur for customer's designs & when those cases were reported for their designs also.

A global was further used that should decommitt moves for the flops in the patched module for clock gating or reset hierarchies if RDC violation occurs.

Now, after global was set, have observed the scenarios where global when set reduces the increased violations in power optimized RTL, & also those cases where enabled global didn't reduce the added violations due to clock gating or reset hierarchies added by PowerPro to have output result as power optimized RTL.

Those scenarios are :

Result 1: There are scenarios where moves on the flops are not killed even though RDC violation is observed on the flops patched by powerPro. Those scenarios are mentioned below:

Scenario 1:

If the moves on the flop is coming from fanin flops (inserted in reset_hier & Powerpro inserted stb hierarchies) and these fanin flops are having RDC violations then global does not have effect on flops (donâĂŹt kill moves on these flops)

Scenario 2:

If the move on the flop is from fanin flops(original rtl & powerPro inserted stb hierarchy)and these fanin flops have RDC violation, then global set, will not disable move.

Result 2: Scenarios, where moves are killed when global is set are mentioned below.

Scenario 1 :

If the moves on the flop is coming from fanin flops inserted in multiple STB hierarchies in different instances of original RTL, and these flops have rdc violation then moves will be killed when global is set.

Scenario 2:

Moves on flops through fanin of chain of flops inserted in PowerPro hierarchies with RDC violation are killed when this global is set.

Also, to understand those scenarios where moves were expected to be killed but they didn't, unit testcases were created to reproduce those cases.

References

[1] Mentor,A Siemens Business Internal Documents

[2] Clock Domain Crossing :
https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja
&uact=8&ved=0ahUKEwizpuu0o97OAhVJ9WMKHRc7DecQFggbMAA&url=
http%3A%2F%2Fwww.eet.bme.hu%2F nagyg%2Fmikroelektronika%2FCadence_CDC
_WP.pdf &usg=AFQjCNG78dcA8OkoPonvmzAqPx0zL1Nmag&sig2=306Pgm3H-cYVsQfLCZs1Ww

[3] Dealing with SoC metastability problems due to Reset Domain Crossing
http://www.embedded.com/design/programming-languages-and-tools/4424093/Dealing-with-SoC-metastability-problems-due-to-Reset-Domain-Crossing-

[4] SoC tool flow techniques for detecting reset domain crossing problems
http://www.embedded.com/design/mcus-processors-and-socs/4433434/2/SoC-tool-flow-techniques-for-detecting-reset-domain-crossing-problems