SmartFusion2 and IGLOO2

Microsemi Separation Verification Tool

User Guide

Table of Contents

Introduction

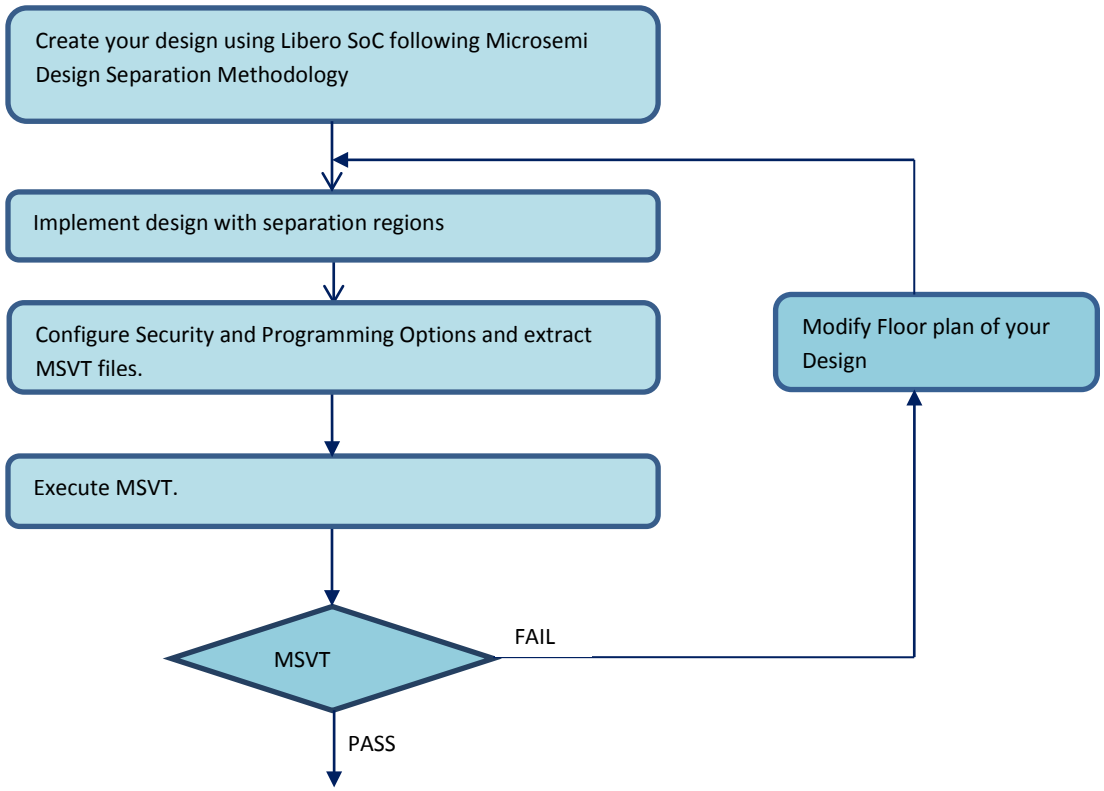
The Microsemi Design Separation methodology provides a method for creating independent critical subsystems required to implement security and safety critical applications on a single FPGA. Microsemi Separation Verification Tool (MSVT) is a standalone tool provided with your Libero installation. It is used to verify that your design meets the requirements of design separation criteria. For more details refer to the Microsemi Design Separation Methodology guide.

MSVT can work on any placed and routed design which has a block requiring a separation from all elements external to the block. The tool works iteratively on every block to be verified. Internal signals and Inter-region Signal (IRS) are verified separately. The tool checks whether the separation criteria is satisfied for each block and corresponding sets of IRS signals.

Your design must adhere to following criteria to implement a security and safety critical system:

- Each Block of your design should be assigned to Separation Region.
- Minimum gap of unused logic clusters between Separation regions depending on your design requirement.
- Each set of inter-block interface signals should be defined as IRS Region.

The flow chart shown in Figure 1 shows the steps of the design separation methodology.



Proceed to Programming

Figure 1. Microsemi Design Separation Methodology

Create Design

A complete design would comprise of sub-systems published in terms of Block Elements. Each Block element is then instantiated in a top-level module. The top-level module is floor planned into separation regions for each sub-system block and overlapping IRS region connecting the Blocks. These IRS regions should also be physically isolated from other IRS regions. Enable the Design Separation Methodology option in the Compile tool as shown in Figure 2. The design is then run through Layout, followed by Timing closure of your design.

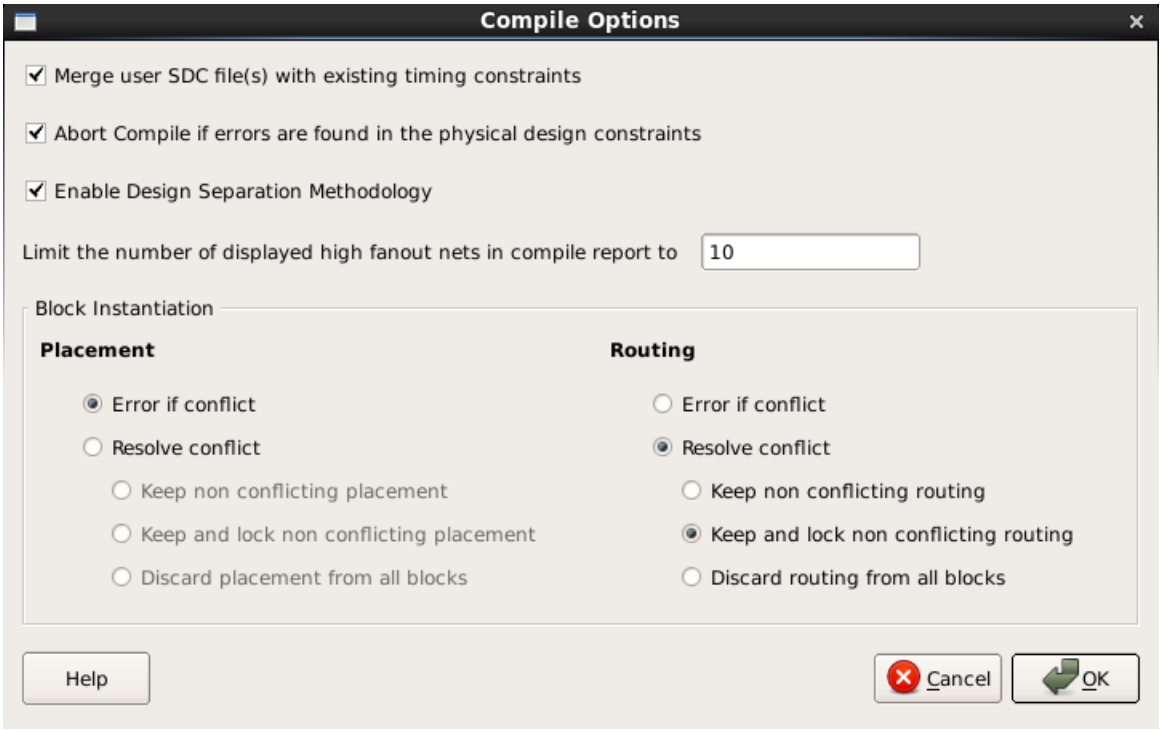


Figure 2. Enabling Design Separation Methodology before Compile

Refer to the Microsemi Design Separation Methodology guide for further details in creating your design.

Extracting MSVT Files

Generate the programming file. This step will also export information for Microsemi Separation Verification Tool (MSVT).

The tool takes as input the design database (the ".dtf" directory within the active project directory) and a parameter file that is generated once per project. The parameter file describes the isolation regions in

the design as well as the inter-region signals between isolation regions. This file is exported to the following location:

<project_path>/designer/<Top_Level_Module>/msvt.param

MSVT.param File

Msvt.param file is an auto-generated file that contains parameters required by MSVT to verify design separation. You can modify the contents of this file according to your requirements.

Below is an example of msvt.param file generated followed by a brief description of each parameter.

Example:
Below is a Top Level view of an example design followed by auto-generated MSVT parameter file

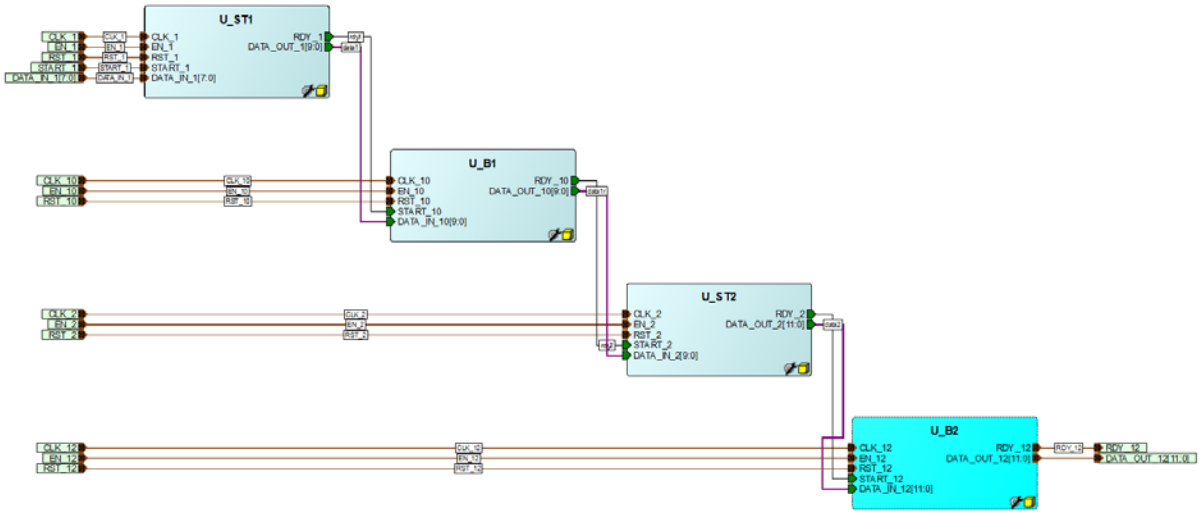


Figure 3a. Top Level Smart Design showing different design blocks

Below is an example parameter file generated for the design above.

```
//*********************************************************************
//
//  This is input parameters file for MSVT Check program
//
//*********************************************************************

DEVICE = M2S050                          // Device Selected
DESIGN = DCT_AAN.msvt                    // Location of Files required for MSVT
VERIFY_BLOCKS = U_B1 U_B2 U_ST2 U_ST1     // List of blocks to be verified using MSVT
                                         // Empty list means all blocks in design will be verified
REQUIRED_SEPARATION = 1                  // specifies the desired number of switch separation
MAX_VIOLATIONS_PER_REPORT_SECTION = 1     // maximum number of violations

// IRS Regions between each pair of blocks. This contains list of inter-block communicating signals.
// The IRS parameter is empty if there are no inter-block communicating signals between a pair of blocks

IRS U_B1 U_ST2 = rdy2 data1r[0] data1r[1] data1r[2] data1r[3] data1r[4] data1r[5]
        data1r[6] data1r[7] data1r[8] data1r[9]
IRS U_ST1 U_B1 = rdy1 data1[0] data1[1] data1[2] data1[3] data1[4] data1[5] data1[6]
        data1[7] data1[8] data1[9]
IRS U_ST2 U_B2 = rdy3 data2[0] data2[1] data2[2] data2[3] data2[4] data2[5] data2[6]
        data2[7] data2[8] data2[9] data2[10] data2[11]
IRS U_B1 U_ST1 =
```

```
IRS U_ST1 U_B2 =
IRS U_B2 U_ST1 =
IRS U_B1 U_B2 =
IRS U_B2 U_B1 =
IRS U_B2 U_ST2 =
IRS U_ST2 U_ST1 =
IRS U_ST1 U_ST2 =
IRS U_ST2 U_B1 =

REGIONS_VERBOSITY = 0                    // to enable verbosity in output Report
-------------------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 3b. Example of auto-generated msvt.param file

	Parameter	Description
1	DEVICE	Name of the Microsemi FPGA device implementing the design.
2	DESIGN	Location of the files required for MSVT. By default it will point to msvt.dtf folder that is auto-generated in your design folder.
3	VERIFY BLOCKS	Contains list of blocks to be verified. By default contains list of all Block names present in the design. You can modify this list and include a subset of Blocks to be audited by MSVT.
4	REQUIRED_SEPARATION	This is the required separation parameter per guideline requirements. Default value is 1.
5	MAX_VIOLATIONS_PER_REPORT_SECTION	This verbosity parameter controls the number of violations that are to be reported in each section. Default value is 1.
6	IRS	Contains list of IRS signal names present in the design. Each IRS statement is comprised of a pair of separated blocks followed by the list of IRS signal names between them.
7	REGIONS_VERBOSITY	This Boolean parameter controls reporting of each routing region and the assigned instances. Default value of this parameter is “0”.

Table 1. Parameters in MSVT.param file

You can modify these parameters to refine your verification criteria. You can specify the blocks you want to verify, the names of each IRS signal, and limit the max number of violations to be reported.

Microsemi Separation Verification Tool (MSVT)

MSVT prints an exhaustive report on each Block and corresponding IRS regions being verified. If any of the Block or IRS signal does not satisfy minimum separation criteria, the tool reports details of affected instances. More information on each section of the report is given in the following sections.

An MSVT failure indicates that the design has not met the design separation criteria and one or more sub-blocks (or signals) are not independent of rest of the system. In this case, identify the instances that cause violations in the MSVT output, and accordingly modify the design floor-plan. Recompile the design to generate a new placed and routed netlist. Verify the modified design using MSVT tool.

Executing MSVT

MSVT is a stand-alone tool present in <Libero_path>/bin64/msvt_check. The tool is executed from the command line. A required argument is –p along with the path to the msvt.param file that is generated from Libero

The command to execute to verify the design using MSVT is:

<Libero_path>/bin64/msvt_check –p <project_path>/designer/<Top_Level_Module>/msvt.param [–o msvt_check.log]

This command will print an exhaustive report into the filename given with the –o argument or to *stdout* if –o is omitted.

On successful completion of this command, you will see a final message of either “MSVT Check Failed” indicating that your design has not met one or more of separation criteria or “MSVT Check succeeded” indicating the design has met all separation criteria.

Analyzing MSVT Report

Successful execution of MSVT will output an exhaustive report with details pertaining to each Block and IRS Regions between each block.

Below is an example of a MSVT generated report for a design which satisfies design separation criteria as specified in msvt.param shown in Figure 3.

```
MSVT Check
Design: DCT_AAN.msvt                    Started: Mon Jun 23 03:07:10 2014

Loading routing information for ECO mode...
After loading previous routing information: Shorts = 0. Open nets = 0.


MSVT Check
Design: DCT_AAN.msvt


Checking IRS connectivity against parameter file
```

```
Analyzing floorplan …
--------------------------------------------------------------------------------

   U_ST2 and U_ST1 : Minimal floorplan separation =  diagonal.
   U_ST2 and U_ST1 : Minimal placement separation =  overlapping.
      U_ST2 at (0,33)
         containing cell U_ST2/INBUF_0/U0/U_IOIN
      U_ST1 at (0,33)
         containing cell U_ST1/IO_0/U0_5/U0/U_IOIN

   U_ST2 and U_B1 : Minimal floorplan separation = 11 clusters.
      U_ST2 at (19,32)
      U_B1 at (19,20)
   U_ST2 and U_B1 : Minimal placement separation =  overlapping.
      U_ST2 at (0,40)
         containing cell U_ST2/INBUF_1/U0/U_IOIN
      U_B1 at (0,40)
         containing cell U_B1/INBUF_1/U0/U_IOIN

   U_ST2 and U_B2 : Minimal floorplan separation = 7 clusters.
      U_ST2 at (19,15)
      U_B2 at (11,15)
   U_ST2 and U_B2 : Minimal placement separation = 7 clusters.
      U_ST2 at (19,15)
         containing cell U_ST2/DCT8AAN2_0/DATA_OUT_2[9]
      U_B2 at (11,15)
         containing cell U_B2/DCT_BUF_12_0/SRL16_a_sr64_0[9]

   U_ST1 and U_B1 : Minimal floorplan separation = 7 clusters.
      U_ST1 at (19,32)
      U_B1 at (11,32)
   U_ST1 and U_B1 : Minimal placement separation =  overlapping.
      U_ST1 at (0,40)
         containing cell U_ST1/IO_0/U0_1/U0/U_IOIN
      U_B1 at (0,40)
         containing cell U_B1/INBUF_1/U0/U_IOIN

   U_ST1 and U_B2 : Minimal floorplan separation = 9 clusters.
      U_ST1 at (0,31)
      U_B2 at (0,21)
   U_ST1 and U_B2 : Minimal placement separation = 11 clusters.
      U_ST1 at (6,31)
         containing cell U_ST1/DCT8AAN1_0/MPU1_ep_1_mulonly_0[20_0]/U0/INST_MACC_IP
      U_B2 at (6,19)
         containing cell U_B2/DCT_BUF_12_0/SRL16_a_sr64_80[7]

   U_B1 and U_B2 : Minimal floorplan separation =  diagonal.
   U_B1 and U_B2 : Minimal placement separation = 19 clusters.
      U_B1 at (0,40)
         containing cell U_B1/INBUF_1/U0/U_IOIN
      U_B2 at (0,20)
         containing cell U_B2/INBUF_2/U0/U_IOIN

Checking internal nets for block U_ST2 …
--------------------------------------------------------------------------------

Checking IRS nets for block U_ST2 …
--------------------------------------------------------------------------------

Propagating IRS nets outgoing from U_ST2 to U_B2
--------------------------------------------------------------------------------

Checking internal nets for block U_ST1 …
--------------------------------------------------------------------------------

Checking IRS nets for block U_ST1 …
--------------------------------------------------------------------------------
```

```
Propagating IRS nets outgoing from U_ST1 to U_ST2 U_B1 U_B2
=============================================================

Propagating IRS nets outgoing from U_ST1 to U_B1
=============================================================

Checking internal nets for block U_B1 …
=============================================================

Checking IRS nets for block U_B1 …
=============================================================

Propagating IRS nets outgoing from U_B1 to U_ST2
=============================================================

Checking internal nets for block U_B2 …
=============================================================

Checking IRS nets for block U_B2 …
=============================================================

Propagating IRS nets outgoing from U_B2 to U_ST2 U_ST1 U_B1
=============================================================

Design has met 1 switches separation requirement
MSVT Check succeeded.
Number of errors: 0
-------------------------------------------------------------
```

Figure 4. MSVT Output report Example

Sections of the MSVT report:

Below is a description on the sections that are present in the MSVT output report along with example of MSVT output:

- Checking IRS connectivity against parameter file

  MSVT checks whether all Inter-region signals are specified as IRS statements in msvt.param file and if the IRS connections specified are consistent with given design netlist. Otherwise, each missing IRS net or the invalid connection is counted as error as is shown in this section.

  Example:
  Msvt.param File:
  Below is an example of msvt.param file in which we haven't mentioned IRS signals rdy1 and rdy2 corresponding to blocks U_ST1-U_B1 and U_B1-U_ST2, respectively. Moreover IRS net rdy3 is an IRS signal between U_ST2 and U_B2 blocks which is incorrectly mentioned in below parameter file as an IRS signal between U_B1 and U_ST2.

```
-------------------------------------------------------------------------------------------------------------------------------
IRS U_B1 U_ST2 = data1r<0> data1r<1> data1r<2> data1r<3> data1r<4> data1r<5> data1r<6>
    data1r<7> data1r<8> data1r<9>  rdy3
IRS U_ST2 U_B2 = data2<11> data2<10> data2<9> data2<8> data2<7> data2<6> data2<5>
    data2<4> data2<3> data2<2> data2<1> data2<0>
IRS U_B2 U_ST2 =
IRS U_ST1 U_B1 = data1<9> data1<8> data1<7> data1<6> data1<5> data1<4> data1<3>
    data1<2> data1<1> data1<0>
```

Figure 4a. Msvt.param file snippet

MSVT Output Report Section:
On executing MSVT with parameter file as an Input , MSVT fails and Reports rdy1 , rdy2, rdy3 as errors as shown below

---

```
Checking IRS connectivity against parameter file
================================================
  Error: IRS net rdy3 has different connectivity than param file
  Error: IRS net rdy1 is not listed in param file
  Error: IRS net rdy2 is not listed in param file
```

Figure 4b. MSVT output Report Section

- Following instance do not belong to any routing region

MSVT will check whether all instances of each block in the design are assigned to a routing region. If all instances are assigned to a separation region then MSVT report will not include this section.

If any Block instance is not assigned to a Routing Region, then MSVT will list the instance name under this section.
Note: This is an information section to identify instances that are not assigned to any region. This will not count as error.

Example:
Design Constraints file:
Let us constrain a design within separation routing regions as given with following physical constraints

---

```
define_region -name UserRegion0 -type inclusive -color 8388736 -route YES -push_place YES 0 93 143 146
define_region -name UserRegion2 -type inclusive -color 12639424 -route YES -push_place YES 240 0 443 62
define_region -name UserRegion3 -type inclusive -color 15780518 -route YES -push_place YES 0 0 143 65
define_region -name UserRegion4 -type inclusive -color 16735838 -route YES -push_place YES 108 111 263 131
define_region -name UserRegion5 -type inclusive -color 975928 -route YES -push_place YES 324 21 371 119
define_region -name UserRegion6 -type inclusive -color 65535 -route YES -push_place YES 96 9 263 47
define_region -name UserRegion1 -type inclusive -color 32896 -route YES -push_place YES 240 96 443 146 276 201 359
206 324 138 347 206
assign_region UserRegion0 U_ST1/DCT8AAN1_0*
assign_region UserRegion0 U_ST1/INBUF_2*
assign_region UserRegion0 U_ST1/INBUF_3*
assign_region UserRegion0 U_ST1/IO_0*
assign_region UserRegion0 U_ST1/CFG0_GND_*

assign_region UserRegion1 U_B1/DCT_BUF_10_0*
assign_region UserRegion1 U_B1/INBUF_2*

assign_region UserRegion2 U_ST2/DCT8AAN2_0*
assign_region UserRegion2 U_ST2/INBUF_2*
assign_region UserRegion2 U_ST2/CFG0_GND*

assign_region UserRegion3 U_B2/DCT_BUF_12_0*
assign_region UserRegion3 U_B2/INBUF_2*
```

```
assign_region UserRegion3 U_B2/IO_0*
assign_region UserRegion3 U_B2/OUTBUF_0*

assign_net_macros UserRegion4 data1<* -include_driver YES
assign_net_macros UserRegion4 rdy1 -include_driver YES
assign_net_macros UserRegion5 data1r<* -include_driver YES
assign_net_macros UserRegion5 rdy2 -include_driver YES
assign_net_macros UserRegion6 data2<* -include_driver YES
assign_net_macros UserRegion6 rdy3 -include_driver YES
```

Figure 5a. PDC constraints of an example design

MSVT Output report:
MSVT will report all instances of the design that are not included in any routing regions as in this the section as shown below.

```
The following instances do not belong to any routing region:
============================================================================
  U_B1/INBUF_0/U0/U_IOIN
  U_B1/INBUF_1/U0/U_IOIN
  U_B2/INBUF_0/U0/U_IOIN
  U_B2/INBUF_1/U0/U_IOIN
  U_ST1/INBUF_0/U0/U_IOIN
  U_ST1/INBUF_1/U0/U_IOIN
```

Figure 5b. MSVT output Report Section

- The following IRS nets are not constrained by any routing region:

MSVT will check whether all IRS nets of the design are assigned to a routing region. If all IRS nets are assigned to an IRS routing region then MSVT report will not display this section.

If any IRS net is not assigned to a Routing Region, then MSVT will show these instances under this section as shown in the example below.

Note: This is an information section to identify nets that are not assigned to IRS region. This will not cause MSVT to fail.

Example:
Design Constraints file:
Let us consider a design with separation routing regions as given with following physical constraints and respective design blocks and IRS regions assigned. In this file we haven't assigned nets rdy1 and rdy2 to a Constrained Routing Region (i.e. we didn't include corresponding driver for these nets as highlighted below):

```
define_region -name UserRegion0 -type inclusive -color 8388736 -route YES -push_place YES 0 93 143 146
define_region -name UserRegion2 -type inclusive -color 12639424 -route YES -push_place YES 240 0 443 62
define_region -name UserRegion3 -type inclusive -color 15780518 -route YES -push_place YES 0 0 143 65
define_region -name UserRegion4 -type inclusive -color 16735838 -route YES -push_place YES 108 111 263 131
define_region -name UserRegion5 -type inclusive -color 975928 -route YES -push_place YES 324 21 371 119
```

```
define_region -name UserRegion6 -type inclusive -color 65535 -route YES -push_place YES 96 9 263 47
define_region -name UserRegion1 -type inclusive -color 32896 -route YES -push_place YES 240 96 443 146 276 201 359
206 324 138 347 206
assign_region UserRegion0 U_ST1/DCT8AAN1_0*
assign_region UserRegion0 U_ST1/INBUF_2*
assign_region UserRegion0 U_ST1/INBUF_3*
assign_region UserRegion0 U_ST1/IO_0*
assign_region UserRegion0 U_ST1/CFG0_GND_*

assign_region UserRegion1 U_B1/DCT_BUF_10_0*
assign_region UserRegion1 U_B1/INBUF_2*

assign_region UserRegion2 U_ST2/DCT8AAN2_0*
assign_region UserRegion2 U_ST2/INBUF_2*
assign_region UserRegion2 U_ST2/CFG0_GND*

assign_region UserRegion3 U_B2/DCT_BUF_12_0*
assign_region UserRegion3 U_B2/INBUF_2*
assign_region UserRegion3 U_B2/IO_0*
assign_region UserRegion3 U_B2/OUTBUF_0*

assign_net_macros UserRegion4 data1<* -include_driver YES
assign_net_macros UserRegion4 rdy1
assign_net_macros UserRegion5 data1r<* -include_driver YES
assign_net_macros UserRegion5 rdy2
assign_net_macros UserRegion6 data2<* -include_driver YES
assign_net_macros UserRegion6 rdy3 -include_driver YES
```

Figure 6a. .PDC constraints of an example design

MSVT Output report:
MSVT will report the nets which are constrained by Routing Region as shown below.

```
The following IRS nets are not constrained by any routing region:
==========================================================================================
  DCT_AAN_0/rdy1
  DCT_AAN_0/rdy2
```

Figure 6b. MSVT output Report Section

- Input signals of the following MACC instances can be observed by failure of config switches in cascade chain

MSVT will check whether all cascaded MACC instances adhere to the separation criteria. If there any such cascaded MACC instances that violate separation criteria, then this section is generated containing a list of MACC instances.

Example:
Design example:
Below is a MVN snap shot of a design in which Top_sign21x18_mult_0, Top_sign21x18_mult_1 and Top_sign21x18_mult_2 are three Math blocks. Top_sign21x18_mult_0 is adjacent to Top_sign21x18_mult_1, whereas Top_sign21x18_mult_2 is separated by Top_sign21x18_mult_1 with one Math cluster.

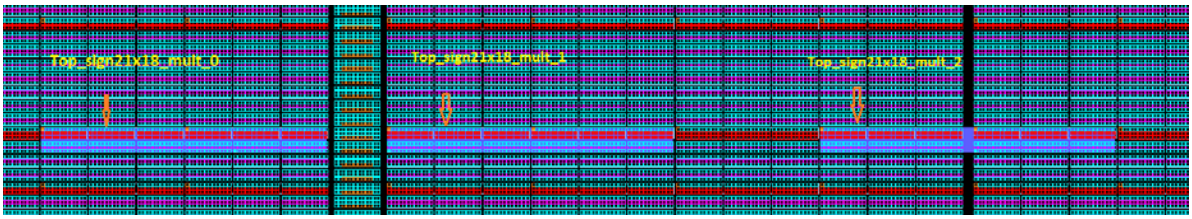


Figure 7a. ChipPlanner view of an example design

MSVT Output Report:

Since Top_sign21x18_mult_0 is adjacent to Top_sign21x18_mult_1, MSVT will fail and reports an error for these MACC instance as shown below. However since Top_sign21x18_mult_2 is separated from Top_sign21x18_mult_1 by at least one MATH cluster, it is not identified as error by MSVT as shown in Report section example below:

Input signals of the following MACC instances can be observed by failure of config switches in cascade chain:
==========

Top_sign21x18_mult_0/sign21x18_mult_0/WideMult_1_0/U0/INST_MACC_IP of block Top_sign21x18_mult_0 can be observed by MACC instance Top_sign21x18_mult_1/sign21x18_mult_0/WideMult_0_0/U0/INST_MACC_IP of block Top_sign21x18_mult_1

Figure 7b. MSVT output Report Section

- Analyzing floorplan

MSVT extracts the separation between each pair of blocks in the design which may help in identifying violations in the floorplan. For each pair of blocks present in the design the following information is generated:
Floorplan Separation: Shows the minimum separation between the respective routing regions in cluster units. No overlap in the either the x or y dimensions of the two regions is indicated as “diagonal”.
Placement Separation: Shows the minimum separation between the actual placements of instances in each block in cluster units. No overlap in the either the x or y dimensions of the placement for instances of the two blocks is indicated as “diagonal”.

This information can be used to identify which Blocks do not have sufficient cluster separation between them leading to MSVT failure.

Note: The coordinates mentioned against blocks is in term of Clusters.

Example:
Design Example:
For example consider blocks U_ST1 and U_B2 separated as shown below in MVN

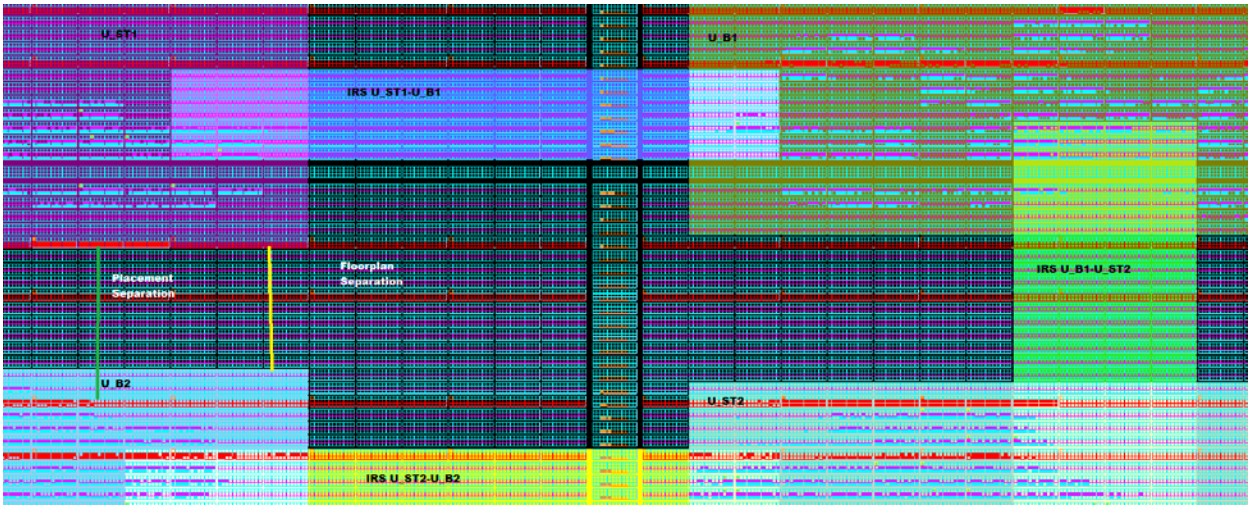


Figure 8a. ChipPlanner view of Design

MSVT Output report:
In the example of report section shown below, the floorplan separation indicates that the regions to which blocks U_ST1 and U_B2 are assigned is separated by 9 Clusters.

Placement separation between instances of blocks U_ST1 and U_B2 actually placed is 11 Clusters.
U_ST1 at (0,31) suggest U_ST1 region spans from 0th Cluster in X- Direction and 31st Cluster in Y-Direction.

Analyzing floorplan …

==========

U_ST1 and U_B2 : Minimal floorplan separation = 9 clusters.

U_ST1 at (0,31)

U_B2 at (0,21)

U_ST1 and U_B2 : Minimal placement separation = 11 clusters.

U_ST1 at (6,31)

containing cell U_ST1/DCT8AAN1_0/MPU1_ep_1_mulonly_0[20_0]/U0/INST_MACC_IP

U_B2 at (6,19) containing cell U_B2/DCT_BUF_12_0/SRL16_a_sr64_80[7]

Figure 8b. MSVT output Report Section

- Checking internal nets for block <Block1>:

MSVT will check that internal nets corresponding to a given block are separated from external nets as per given separation criteria.

This section will be empty if your design does not have any net violating separation criteria. If any of the nets of the design fails to satisfy separation criteria, then

information related to the violating net will be displayed in this Section and MSVT will treat this as an Error.

Example:
MSVT Output Report:
The example below shows internal net “U_ST2/DCT8AAN2_0/un1_cp_0[9] “ of Block U_ST2 is failing separation criteria and the net can access untrusted net “U_B2/DCT_BUF_12_0/DATA_OUT_12_6_bm[9] “ through switches present at coordinates mentioned below :

```
---------------------------------------------------------------------------------------------------------------------------------------------
        Checking internal nets for block U_ST2 ...
        ================================================================
    Net U_ST2/DCT8AAN2_0/un1_cp_0[9] can be observed by net U_B2/DCT_BUF_12_0/DATA_OUT_12_6_bm[9] (cell
U_B2/DCT_BUF_12_0/DATA_OUT_12_6_ns[9]) , with cluster separation (12,2) by failure of the following 2 switches:
      Switch at (982,1307) failing from OFF to ON
      The path also driven by power via existing ON switch at (985,1306)
      Switch at (391,1305) failing from OFF to ON
      The path also driven by signal  U_B2/DCT_BUF_12_0/DATA_OUT_12_6_bm[9] via existing ON switch at (392,1300)
      Using existing ON switch at (398,1202) used in routing of U_B2/DCT_BUF_12_0/DATA_OUT_12_6_bm[9]
      Using existing ON switch at (399,1253) used in routing of U_B2/DCT_BUF_12_0/DATA_OUT_12_6_bm[9]
---------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 9a. MSVT output Report Section

Routing Details:
We can view the Routing details of the untrusted net (“U_B2/DCT_BUF_12_0/DATA_OUT_12_6_bm[9]“ in above example) by viewing routing details from ChipPlanner till the point where Switch is in ON state

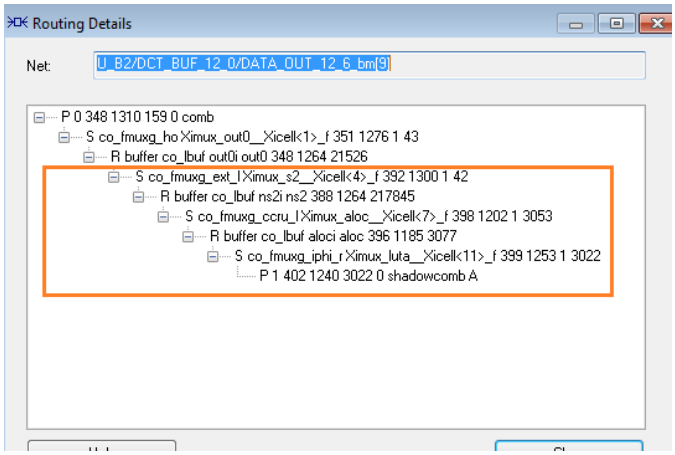


Figure 9b. Routing Details of corresponding net from MVN

- Checking IRS nets for block <Block1>:

MSVT will check that IRS nets corresponding to a given block are separated from external nets as per given separation criteria.

This section will be empty if your design does not have any net violating separation criteria. If any of the nets of the design fails to satisfy separation criteria, then information related to the violating net will be displayed in this section and counted as error.

Example:
MSVT Output report:
Below is an example of this section in which IRS net RDY_1 of block Top_DCT8AAN1_0 is being observed by an external net corresponding to another untrusted Logic

```
Checking IRS nets for block Top_DCT8AAN1_0 ...
================================================================
The following outgoing IRS nets have 0 switches separation since they are connected directly to at least one untrusted logic:
  Top_DCT8AAN1_0_RDY_1 (cell RCLKINT_0/U0)
```

Figure 10a. MSVT output Report Section

- Propagating IRS nets outgoing from <Block1> to <Block2>:

MSVT will check that IRS nets corresponding to a given block are separated from external nets as per given separation criteria.
This section will be empty if your design does not have any net violating separation criteria. If any of the nets of the design fails to satisfy separation criteria, then information related to the violating net will be displayed in this section and counted as error.

Example:
MSVT Output Report:
The example below shows IRS net data1r<8> connecting U_B1 and U_ST2 instances is failing to meet separation criteria. This net can be observed by external net "U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9] " of block "U_B2".

```
Propagating IRS nets outgoing from U_B1 to U_ST2
================================================================
  Net data1r<8> can be observed by net U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9] (cell U_B2/DCT_BUF_12_0/DATA_OUT_12_18_RNI4L361[9]) , with cluster separation (12,0) by failure of the following 2 switches:
  Switch at (1129,1224) failing from OFF to ON
  The path also driven by power via existing ON switch at (1129,1227)
  Switch at (551,1220) failing from OFF to ON
  The path also driven by signal  U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9] via existing ON switch at (552,1210)
  Using existing ON switch at (550,1290) used in routing of U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9]
  Using existing ON switch at (548,1265) used in routing of U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9]
  Using existing ON switch at (545,1320) used in routing of U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9]
  Using existing ON switch at (545,1342) used in routing of U_B2/DCT_BUF_12_0/DATA_OUT_12_18[9]
```

Figure 11a. MSVT output Report Section

Above report indicates that an IRS signal through net data1r<8> between blocks U_B1 to U_ST2 is not separated by the number of required switches specified in DESIGN_SEPARATION parameter from another net DATA_OUT_12_18[9] which is a part of U_B2 block.

If parameter REGIONS_VERBOSITY is set to '1', then MSVT will output additional information related to the floorplan. Following two sections are generated only if REGIONS_VERBOSITY = 1. These sections provide additional information related to design and do not count as errors.

- The following region constraints are associated with block <Block_Name>

  MSVT will describes in detail all the block instances that are associated with a given separation region.

  In case a separation region is a rectilinear region, then each sub-rectangular region is analyzed for block instances.

  Example:
  MSVT Output Report:
  The example below shows such a section where block U_ST2 is part of a rectilinear separation region. The report shows two region constraints associated with U_ST2 block, each with coordinates of two sub-rectangular regions corresponding to the rectilinear region.

```
The following region constraints are associated with block U_ST2:
================================================================================
(INCLUSIVE REGION
        (RECT 1212 1738 2468 3792)
        (CELLS
                U_ST2/CFG0_GND_INST
                U_ST2/DCT8AAN2_0/DATA_OUT_Z [0]
                                .
                .
                .
                .
                .
                U_ST2/DCT8AAN2_0/un8_bp_m_am[9]
                U_ST2/INBUF_2/U0/U_IOIN
        )
)

The following region constraints are associated with block U_ST2:
================================================================================
(INCLUSIVE REGION
        (RECT 588 2844 1892 3713)
        (CELLS
                U_ST2/DCT8AAN2_0/RDY_RNO_0
                U_ST2/DCT8AAN2_0/UU_COUN0_ad1_4[0]
                U_ST2/DCT8AAN2_0/UU_COUN0_ad1_4_rep1[1]
                .
                .
                .
                .
                U_ST2/DCT8AAN2_0/di[9]
        )
)

The following region constraints are associated with block U_ST1:
================================================================================
( INCLUSIVE REGION
        ( RECT 4 0 1068 1896)
        ( CELLS
                U_ST1/IO_0/U0_0/U0/U_IOIN
```

```
                    .
                    .
                    .
                    U_ST1/INBUF_3/U0/U_IOIN
            )
)

The following region constraints are associated with block U_B1:
================================================================================
( INCLUSIVE REGION
            ( RECT 4 2449 1116 3792)
            ( CELLS
                    U_B1/DCT_BUF_0/DATA_OUT_BUF_4[0]
                    .                       .
                    .
                    U_B1/DCT_BUF_0/cycles_s[5]
            )
)

The following region constraints are associated with block U_B2:
================================================================================
( INCLUSIVE REGION
            ( RECT 1548 0 2472 1738)
            ( CELLS
                    U_B2/OUTBUF_0/U0/U_IOTRI
                    .
                    .
                    .
                    ]
                    U_B2/DCT_BUF12_0/cycles_s[5]
            )
)
--------------------------------------------------------------------------------------------------------------------------------
```

Figure 12a. MSVT output Report Section

- The following are empty regions:

  The section of MSVT output report shows the regions which are defined as Empty Regions. These regions should also be defined as Routing Regions. This section is generated only if there are such empty routing regions in the design.

  Example:

  Below is ChipPlanner view of the input design which is having three Empty Routing Regions defined:

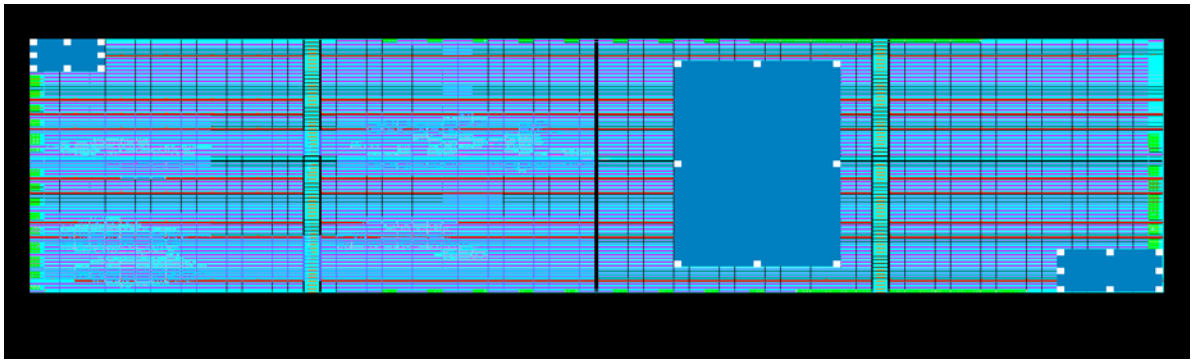


Figure 13a. Chip Planner view of Example design

MSVT Output:

MSVT will report the three Empty Routing Regions in its report as shown below

```
The following are empty region(s):
================================================================================
( EXCLUSIVE REGION
        ( RECT 1980 553 2508 4898)
)
( EXCLUSIVE REGION
        ( RECT 3140 0 3480 948)
)
( EXCLUSIVE REGION
        ( RECT 0 4661 244 5372)
)
```

Figure 13b. MSVT output Report Section

Report Conclusion

On successful completion of MSVT, the report will show a final message as shown below. Number of Errors indicates the total number of errors reported by MSVT in the output report.

```
Design has met 1 switches separation requirement
MSVT Check succeeded.
Number of errors: 0
```

On a MSVT separation criteria failure, the report will show final message as:

```
Design failed for 10 switches separation requirement
MSVT Check failed.
Number of errors: 7
```

You can now program your FPGA with generated programming file.

Reference Documents

Please refer to following documents to implement a design using Design Separation Methodology:

1) "Microsemi Design Separation Methodology"
2) "SmartFusion2 ad IGLOO2 Block Flow User's Guide"
3) "SmartFusion2 and IGLOO2 SmartTime, I/O Editor and ChipPlanner User's Guide"

A – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 408.643.6913

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

5-02-00582-0/09.14