Frequently Asked Questions
Synplify Synthesis

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

© 2017 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

Contents

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 2.0

The following is a summary of the changes in revision 2.0 of this document.

- All the Actel links were updated with the Microsemi links.
- All instances of IDE are removed from the licensing section. For more information, see Licensing Download Installation, page 3
- FAQ 3.9 was added.For more information, see Is Synplify Pro Synthesis tool supported in all the Libero licenses?, page 4.
- FAQ 4.1 was updated.For more information, see Warning: Top entity isn't set yet!, page 5
- FAQ 4.4 was updated.For more information, see Error: The profile for tool Synplify is interactive and you are running in batch mode: this tool cannot be invoked, page 6
- FAQ 5.5 was updated.For more information, see How can I add an attribute in Synplify?, page 8

1.2 Revision 1.0

Revision 1.0 was the first publication of this document.

2 Introduction to Synopsys Synplify

This section answers the basic queries related to Synopsys Synplify tool.

2.1 What does Synplify do?

The Synplify and Synplify Pro products are logic synthesis tools for field programmable gate arrays (FPGAs) and CPLDs complex programmable logic devices CPLDs. The Synplify Pro tool is an advanced version of the Synplify tool, with many additional features for managing and optimizing complex FPGAs. Some additional features available in Synplify Pro are FSM explorer, FSM viewer, Register re timing, and gated clock conversion.

These tools accept high-level input written in industry-standard hardware description languages (Verilog and VHDL), and using the Synplicity behavior extracting synthesis technology (BEST) algorithms, they convert the designs into small and high performance design netlists for popular technology vendors. They write VHDL and Verilog netlists after synthesis, which can be simulated in order to verify functionality.

2.2 Which HDL language does Synplify support?

Verilog 95, Verilog 2001, System Verilog IEEE (P1800) standard, VHDL 2008, and VHDL 93 are supported in Synplify. For information on different language constructs, see https://www.microsemi.com/document-portal/doc_download/136263-synopsys-fpga-synthesis-synplify-pro-me-j2015-03msp1-2-reference-for-libero-soc-v11-7.

2.3 Will Synplify accept manual instantiations of Microsemi macros?

Yes. Synplify contains built-in macro libraries for all of the Microsemi's hard macros including logic gates, counters, flip flops, and I/Os. You can manually instantiate these macros in your Verilog and VHDL designs, and Synplify passes them through to the output netlist.

2.4 How does Synplify work with Microsemi tools?

The Synopsys Synplify Pro ME (Microsemi Edition) synthesis tool is integrated into the Libero, that enables you to target and fully optimize your HDL design for any Microsemi device. As with all other Libero tools, you can launch Synplify Pro ME directly from the Libero Project Manager.

Synplify Pro ME is the standard offering in Libero editions. Synplify Pro ME is launched by invoking the executable specific in the Libero tool profile.

3 Licensing Download Installation

This section answers the queries related to the license installing and download procedure.

3.1 Where can I download the latest Synplify release?

Synplify is a part Libero download and the standalone installation link is as follows.

https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#downloads.

3.2 Which version of Synplify is released with the latest Libero?

For the list Synplify versions released with Libero, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/libero-soc#device-support.

3.3 How do I upgrade to the latest version of Synplify and use it in the Libero Project Manager?

Download and install the latest version of Synplify from the Microsemi or Synopsys website, and change the synthesis settings in the Libero Project Manager tool profile from the Libero Project->Profiles Menu.

3.4 Do I need a separate license to run Synplify in Libero?

No. All Libero licenses except for the Libero-SA (standalone) license includes a license for the Synplify software.

3.5 Where and how do I get the license for Synplify?

To apply for a free license, see http://www.microsemi.com/products/fpga-soc/design-resources/licensing and click the Software Licenses and Registration System link. Input the required information, including the volume ID of your C drive. Make sure to apply with your C drive, even if that is not the drive you intend to install the software on. For paid licenses, contact your local Microsemi Sales Office.

3.6 Why can’t I run Synplify in batch mode? What license does it require?

From a command prompt, go to the directory where the project files are located and type the following.

For Libero IDE: synplify_pro -batch -licensetype synplifypro_actel -log synpl.log TopCoreEDAC_syn.prj

For Libero SoC: synplify_pro -batch -licensetype synplifypro_actel -log synpl.log asdasd_syn.tcl

You must have a Platinum license to run Synplify in batch mode.To purchase a Platinum license, contact the local Microsemi sales representative

3.7 Why is my Synplify license not working?

Following are the steps to check for the proper functioning of the license.

1. Check if the license has expired.
2. Check if the LM_LICENSE_FILE is set correctly as a windows user environment variable which points to the location of the Libero License.dat file.
3. Check if the Libero IDE tool profile set to Synplify Pro?
4. Check if the Synplify license feature is enabled in your license file.
5. Look for the “synplifypro_actel” feature line in license.dat and make sure the HostID is correct for the computer you are using.

```
INCREMENT synplifypro_actel snpslmd 2016.09 21-nov-2017 uncounted \
     4E4905A56595B143FFF4 VENDOR_STRING=^1+S \
     HOSTID=DISK_SERIAL_NUM=ec4e7c14 ISSUED=21-nov-2016 ck=232 \
     SN=TK:4878-0:1009744:181759 START=21-nov-2016
```

3.8 Can I use the Synplify license obtained from Microsemi to run any version of Synplify?

No, if you received a Synplify license from Microsemi, you will only be able to run Synplify ME.

3.9 Is Synplify Pro Synthesis tool supported in all the Libero licenses?

Synplify Pro Synthesis tool is not supported in all the License types. For more information about licensing, see https://www.microsemi.com/products/fpga-soc/design-resources/licensing#overview.

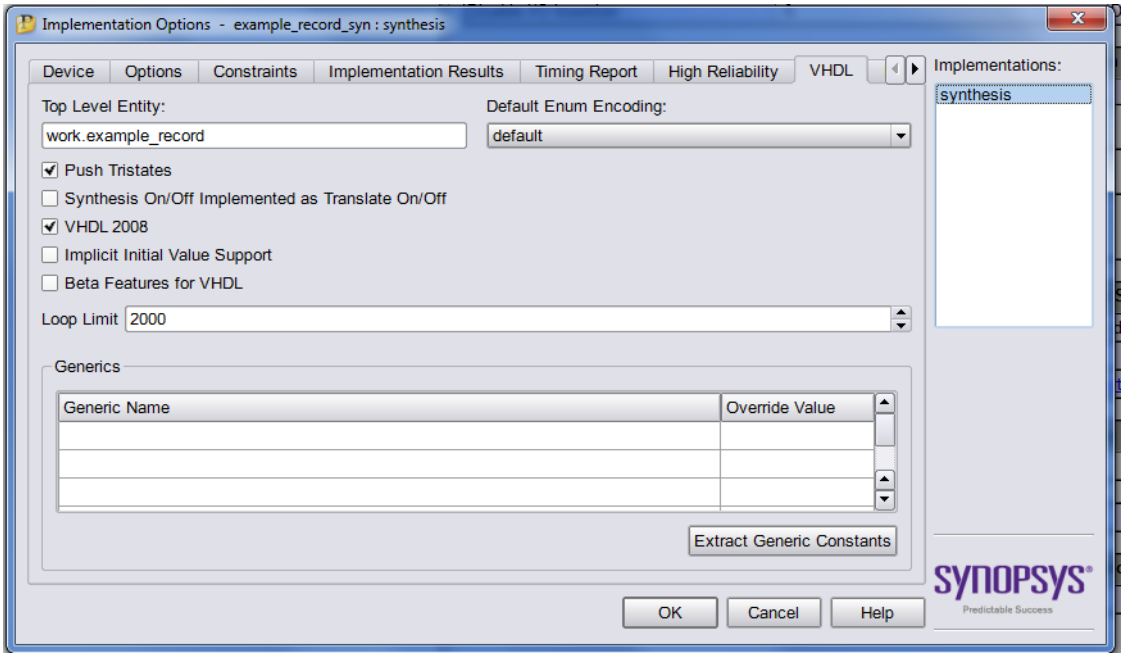
4 Warnings/Error Messages

This section provides information about various error messages that appear during the installation procedure.

4.1 Warning: Top entity isn't set yet!

Synplify could not identify the top entity in your design, due to design complexity. You need to manually specify the top entity name in Synplify implementation options. The following figure shows an example.

Figure 1 • **Example To Specify Top Entity Name**



4.2 Warnings on Register Pruning

Synplify optimizes the design by pruning unused/duplicate registers/nets/blocks. You can manually control the amount of auto optimization by applying the following directives.

- *syn_keep ensures that if a wire is kept during synthesis and hat there are no optimizations cross the wire. This directive is usually used to break unwanted optimizations and to ensure manually created replications. It works only on nets and combinational logic.
- *syn_preserve ensures that registers are not optimized away.
- *syn_noprune ensures that a black box is not optimized away when its outputs are unused (that is, when its outputs do not drive any logic).

For more information about optimization control and Synplify documents, see http://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

4.3 @W: FP101 |The design has 8 instantiated global buffers but allowed is only 6

@W: FP103— User can use syn_global_buffers to increase the allowed global clock buffers to maximumof18.

The warnings are created because Synplify identified more than 6 global macros instantiated in the design. The default maximum number of global nets allowed in Synplify is currently set to 6. So when the tool tries to use more than 6 for this design, it generates an error. You can manually increase the default limit to 8 (up to 18 in IGLOO/e, ProASIC3/E, and Fusion and upto 8 and 16 depending on the SmartFusion2 and IGLOO2 device) by adding a synthesis attribute called syn_global_buffers.

For example:

```
module top (clk1, clk2, d1, d2, q1, q2, reset) /* synthesis
syn_global_buffers = 8 */;
```

……

or

…

```
architecture behave of top is
attribute syn_global_buffers : integer;
attribute syn_global_buffers of behave : architecture is 8;
```

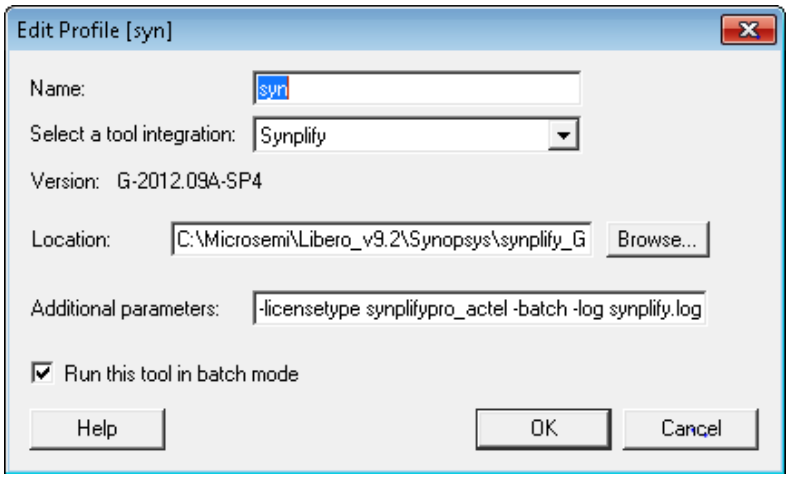
……

For more information, see http://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

4.4 Error: The profile for tool Synplify is interactive and you are running in batch mode: this tool cannot be invoked

You must have a Platinum license to run Synplify in batch mode. Contact the local Microsemi sales representative to purchase a Platinum license. You must ensure that the Libero Synthesis tool profile is configured to launch Synplify in batch mode if you are invoking Synplify from within Libero instead of directly from the command prompt. The following figure shows how to invoke Synplify from within Libero.

Figure 2 • **Example to Invoke Synplify from Within Libero**



4.5 @E: CG103: "C:\PATH\code.vhd":12:13:12:13|Expecting expression

@E: CD488: "C:\PATH\code.vhd":14:11:14:11—EOF in string literal

A comment following anything other than a semicolon or a new line is illegal in VHDL. Two hyphens mark the start of a comment, which is ignored by the VHDL compiler. A comment can be on a separate line or at the end of a line of VHDL code. The error is due to comments in some other part of the VHDL code.

4.6 @E: Internal Error in m_proasic.exe

This is not an expected tool behavior. For more information, contact Synopsys Synplify support team or Microsemi Tech Support if you do not have a Synopsys Support Account.

4.7 Why has my logic block disappeared after synthesis?

Synplify optimizes away any logic block that does not have any external output port.

5 Attributes/Directives

This section answers the queries related to attributes and directives.

5.1 How do I turn off automatic clock buffer usage in Synplify?

To turn off automatic clock buffering for nets or specific input ports use the syn_noclockbuf attribute. Set the Boolean value to one or true to turn off automatic clock buffering.

You can attach this attribute to a hard architecture or module whose hierarchy will not be dissolved during optimization of a port, or net.

For more information about usage of the attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.2 Which attribute is used for preserving registers?

syn_preserve directive is used for preserving registers.

For more information about this attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.3 Does syn_radhardlevel attribute support IGLOO and Fusion families?

No. syn_radhardlevel attribute is not supported in IGLOO and Fusion families. For more information about the TMR setting of the syn_radhardlevel, see TMR Usage, page 14.

5.4 How do I disable serial optimization in Synplify?

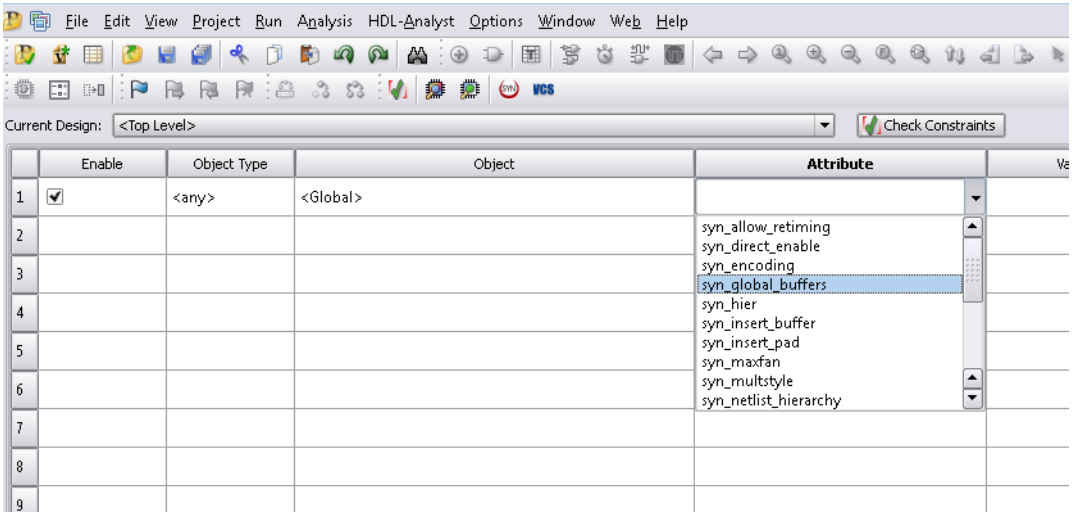
Use syn_preserve directive to disable serial optimization in Synplify.

5.5 How can I add an attribute in Synplify?

Following are the steps to add an attribute in Synplify.

1. Launch Synplify from the Libero Project Manager.
2. Click on File -> New -> FPGA Design Constraints.
3. Click the Attributes tab at the bottom of the spreadsheet.
4. Double-click on any of the attribute cells in the spreadsheet. You should see a pull-down menu with many attributes listed. Select any of them, and fill in the required fields accordingly as shown in the following figure.
5. Save the files and close the Scope Editor after completing the task.

Figure 3 • **Selecting Attributes from the Drop-down cell**



5.6 How do I insert a clock buffer in my design?

Use syn_insert_buffer attribute to insert a clock buffer. The synthesis tool inserts a clock buffer according to the vendor-specific values you specify. The attribute can be applied on instances.

For more information about the usage of the attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.7 How do I increase the number of global clock buffers used in my design?

Use syn_global_buffers attribute in the SCOPE to specify the number of global buffers to be used in a design. It is an integer between 0 and 18. For more information, about this attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.8 Is there any way to preserve my logic if the output ports are not used in my design?

Use syn_noprune attribute to preserve the logic if the output ports are not used in the design.

For example: module syn_noprune (a,b,c,d,x,y); /* synthesis syn_noprune=1 */;

For more information, about this attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.9 Why is synthesis optimizing my high fanout net to buffered clock?

Use syn_maxfan to override the default (global) fanout guide for an individual input port, net, or register output. Set the default fanout guide for a design through the device panel on the Implementation Options dialog box or with the set_option -fanout_limit command in the project file. Use the syn_maxfan attribute to specify a different (local) value for individual I/Os.

For more information, about this attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.10 How do I use the syn_encoding attribute for an FSM design?

The syn_encoding attribute overrides the default FSM compiler encoding for a state machine. This attribute takes effect only when FSM compiler is enabled. Use syn_encoding when you want to disable the FSM compiler globally but there are a select number of state registers in your design that you want extracted. In this case, use this attribute with the syn_state_machine directive on for just those specific registers.

For more information, about this attribute, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

5.11 Why Synplify generates a netlist that exceeds the maximum fanout of device, causing the netlist to fail compile?

A CC macro (available for Antifuse families) is a flip-flop element built using two C-cells. A net driving the CLK or CLR port of a CC macro in reality is driving two cells. The hard fan-out limit on certain nets does not achieve the desired results because it fails to take this net doubling effect into account.

Include the syn_maxfan attribute in the RTL code to force Synplify to generate a valid netlist. Reduce the max fanout limit value by one for every CC macro driven by the net. For example: Set the syn_maxfan limit to 12 for a net that is driving CC macros to keep the fanout at 24 or less.

6 RAM Inference

This section answers the queries related to the RAM inference Synplify support for Microsemi product families.

6.1 Which Microsemi families do Synplify support for RAM inference?

Synplify supports the Microsemi ProASIC, ProASIC PLUS, ProASIC3, SmartFusion2, IGLOO2, and RTG4 families in generating both single- and dual-port RAMs.

6.2 Is RAM inference ON by default?

Yes. The synthesis tool automatically infers RAM.

6.3 How can I turn off RAM inference in Synplify?

Use syn_ramstyle attribute and set its value to registers.

For more information, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

6.4 How do I make Synplify infer embedded RAM/ROM?

Use syn_ramstyle attribute and set its value to block_ram or LSRAM and USRAM for SmartFusion2 and IGLOO2 devices.

For more information, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

6.5 I cannot compile an existing design in a newer version of designer?

There could be possible RAM/PLL configuration change. Regenerate your RAM/PLL by opening the core configuration options from the Catalog in the Libero Project Manager, and resynthesize/ compile/layout.

7 Area/Quality of Results

This section answers the queries related to the area or quality usage for Synplify.

7.1 Why does area usage increase in the new version of Synplify?

Synplify is designed to achieve better timing results in every new version. Unfortunately, the trade-off is often an area increase.

If the timing requirement is achieved for the design, and the remaining task is to fit the design in a specific die, following are the methods:

1. Increase Fanout limit to reduce buffer replication
2. Change global frequency settings to relax the timing requirement
3. Turn on resource sharing (design specific) to optimize the design

7.2 What kind of area improvement technique is available in Synplify?

Following are the area improvement techniques available in Synplify.

1. Increase the fanout limit when you set the implementation options. A higher limit means less replicated logic and fewer buffers inserted during synthesis, and a consequently smaller area. In addition, as place-and-route tools typically buffer high fanout nets, there is no need for excessive buffering during synthesis.
2. Check the Resource Sharing option when you set implementation options. With this option checked, the software shares hardware resources like adders, multipliers, and counters wherever possible, and minimizes area.
3. For designs with large FSMs, use the gray or sequential encoding styles, because they typically use the smallest area.
4. If you are mapping into a CPLD and do not meet area requirements, set the default encoding style for FSMs to sequential instead of one hot.

7.3 How do I disable area optimization?

The optimization for timing is often under the expense of area. There is no specific way to disable area optimization. Following are the options to improve timing and thereby increase area utilization.

1. Enable re-timing option
2. Enable Pipelining option
3. Use realistic design constraints, about 10 to 15 percent of the real goal
4. Select a balanced fanout constraint

For more information about the optimization for timing, see https://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

7.4 How do I disable sequential optimization?

There is no explicit button or checkbox to disable sequential optimization. This is because there are different types of sequential optimizations that are performed by Synplify.

For more information about the options for disabling optimization, see http://www.microsemi.com/products/fpga-soc/design-resources/design-software/synplify-pro-me#documents.

For example, following are some options to disable optimization.

- Disable the FSM compiler
- Use the syn_preserve directive to keep registers in certain cases

Note: The Project Manager overwrites the Synthesis PRJ file every time you invoke synthesis when choosing this option.)

8 TMR Usage

This section answers the queries related to the TMR usage for Synplify.

8.1 Which family is TMR supported through Synplify?

It is supported on Microsemi ProASIC3/E, SmartFusion2, and IGLOO2 devices as well as Microsemi's Radiation Tolerant (RT) and Radiation Hardened (RH) devices. You can also get the triple module redundancy (TMR) setting to work for Microsemi's older Antifuse Device Families. However it is not supported in the commercial AX device family.

Note: In Microsemi's RTAX device family, better TMR support is available through hardware itself. For Axcelerator RT devices, the TMR is built into the silicon making soft TMR via the Synthesis tool unnecessary for sequential logic.

8.2 Why is TMR macro working in SX, but not in AX family?

There is no software TMR support in Synplify synthesis for the commercial Axcelerator family, but it is available for the SX family. If you are using RTAXS devices, the TMR is built-in (for the sequential flip-flops) in the hardware/device.

8.3 How can I enable TMR for a SX-A device?

For the SX-A device family, in the Synplify software you need to manually import the file found in the Libero IDE Installation Folder, such as:

C:\Microsemi\Libero_v9.2\Synopsys\synplify_G201209ASP4\lib\actel\tmr.vhd.

Note: The order of the files in the Synplify project is important and the top level file should be at the bottom. You can click and hold the top level file in the Synplify project and drag it below the tmr.vhd file.

9 Miscellaneous

This section answers the miscellaneous queries related to Synplify.

9.1 Which version of Synplify supports nano products?

All versions of Synplify after Synplify 9.6 A support nano products.

9.2 Which version of Synplify provides RTAX-DSP support?

All versions included with Libero IDE v8.6 and later provide RTAX-DSP support.

9.3 How do I create an IP core with the HDL files I have?

Create an EDIF net-list without IO buffer insertion. This EDIF net-list is sent to the user as an IP. The user must treat this as a black box and include it in the design.

9.4 Nano devices have only four global clock networks. How do I set this constraint?

Use the attribute /* synthesis syn_global_buffers = 4*/ to set the constraint.

9.5 Why am I not seeing my new port list even after I updated the netlist?

Although the new port was added in the design, the net-list did not add a buffer to the port since there was no logic in the design which involves the port. Ports not associated with any logic in the design are not shown.

9.6 Why is Synplify not using Global for Set/Reset signals?

Synplify treats set/reset signals differently from clocks. Synplify global promotion always gives first priority to clock signals, even if some set/reset signals have higher fanout than clock nets. Manually instantiate a clkbuf to ensure that the set/reset signal is global, if you want to use global network for these signals.

9.7 Why does Synplify write out SDC clock constraints even for auto-constraints?

This is the default behavior in Synplify and cannot be changed. However, you can control the SDC auto-constraints by manually modifying or removing the unwanted constraints.

9.8 Why is my internal tristate logic not synthesized correctly?

Microsemi devices do not support internal tri-state buffers. If Synplify does not correctly remap internal tri-state signals, all internal tri-states must be manually mapped to a MUX.