DesignWare® Cores SerDes PHY Temperature Sensor Procedure

Application Note

Document Version: 1.00a
September 2021

Synopsys, Inc.

www.synopsys.com

Contents

Revision History

The following table lists the revision history of operation and temperature measuring procedures using the Synopsys Temperature Sensor from release to release.

Note

- Links and references to section, table, figure, and page numbers in this table are only assured to be valid for the version in which the change is made.
- In some instances, documentation-only updates occur. The DesignWare IP product information (http://www.designware.com) has the latest documentation.

Date	Doc Version	Description
September 2021	1.00a	Initial version

Preface

This Application Note describes the Synopsys DesignWare® Cores Serdes PHY Temperature Sensor.

Application Note Organization

This Application Note is organized as follows:

- Chapter 1, "Implementation", describes the circuit implementation as well as the registers required to implement the procedure.
- Chapter 2, "Procedure and Measurements", details the temperature sensor procedure implementations.

Related Documentation

DesignWare® Cores Multi-Protocol 16G PHY Databook is packaged with the product. It is also available for download from DWDL.

Web Resources

- DesignWare IP product information: https://www.synopsys.com/designware-ip.html
- Your custom DesignWare IP page: https://www.synopsys.com/dw/mydesignware.php
- Documentation through SolvNetPlus: https://solvnetplus.synopsys.com (Synopsys password required)
- Synopsys Common Licensing (SCL): https://www.synopsys.com/keys

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

Customer Support

To obtain support for your product, contact Support Center using one of the following methods:

- For *fastest response*, enter a case through SolvNetPlus:

  a. https://solvnetplus.synopsys.com

  > **Note** SolvNetPlus does not support Internet Explorer. Use a supported browser such as Microsoft Edge, Google Chrome, Mozilla Firefox, or Apple Safari.

  b. Click the **Cases** menu and then click **Create a New Case** (below the list of cases).

  c. Complete the mandatory fields that are marked with an asterisk and click **Save**.

  Make sure you include the following:

  - **Product L1:** DesignWare Cores
  - **Product L2:** SerDes PHY Temperature Sensor Procedure

  For more information about general usage information, refer to the following article in SolvNetPlus:

  https://solvnetplus.synopsys.com/s/article/SolvNetPlus-Usage-Help-Resources

- Or, send an e-mail message to support_center@synopsys.com (your e-mail will be queued and manually routed to the correct support engineer on a first-come, first-served basis):
  - Include the Product L1 and Product L2 names, process, and Version number in your e-mail so that it is routed correctly.
  - For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood.
- Or, telephone your local support center:
  - North America:

    Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.
  - All other countries:

    https://www.synopsys.com/support/global-support-centers.html

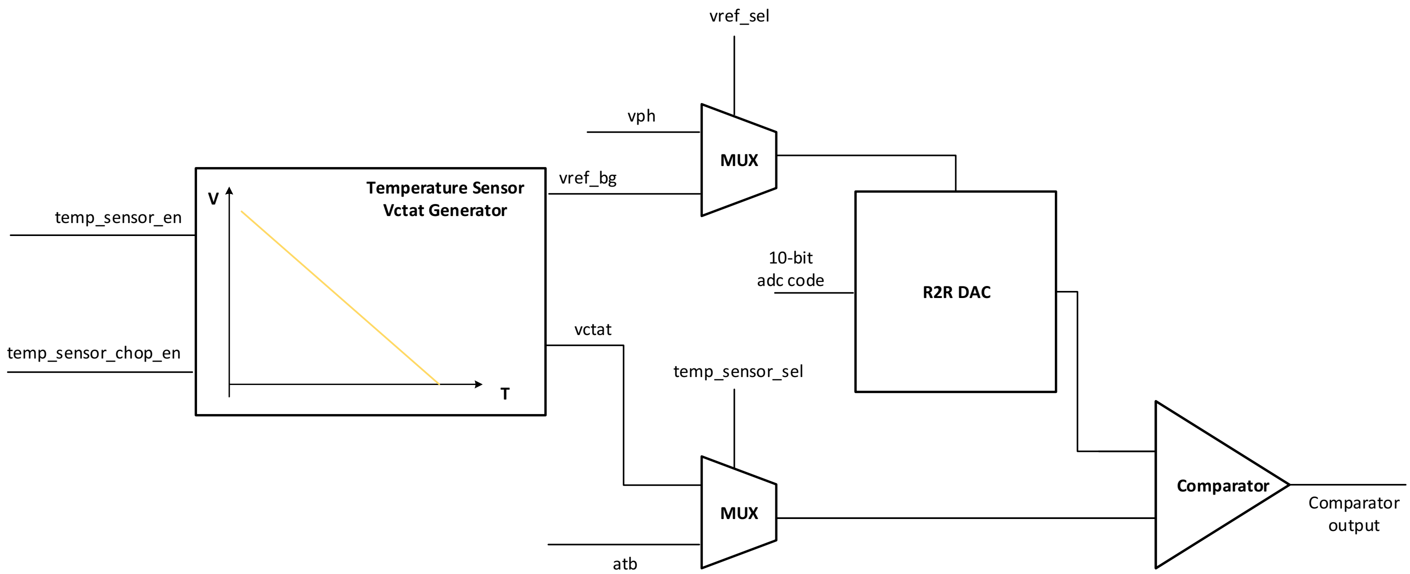
1
Implementation

This chapter includes the following sections:

1.1 Circuit Implementation

A voltage (Vctat) that is complimentary to absolute temperature is generated based on existing bandgap architecture. This voltage is sent to the ADC via a series of register writes that enable various MUXs. This voltage is then digitized into a 10-bit code using the exist ADC function built into the RTUNE block. This code is used to calculate the temperature.

Figure 1-1 Temperature Sensor Block Diagram



1.2 Register Access for Temperature Sensor Procedure

This section provides a summary and a list of all the Register access needed to enable and implement the complete temperature sensor procedure.

1.2.1 Register Writes to Enable the Temperature Sensor

Table 1-1 describes the registers required to enable the Temperature Sensor.

Table 1-1 Register Writes to Enable the Temperature Sensor

Register Name	Read/ Write	Value to be written	Comments
SUP_ANA_BG.TEMP_SENSOR_EN	W	1	Enable temperature sensor circuit (On by default)
SUP_ANA_RTUNE_CTRL.TEMP_SENSOR_SEL	W	1	Select the temperature sensor Vctat voltage for ADC measurement
SUP_ANA_RTUNE_CTRL.VREF_SEL	W	1	Select the Vref_bg as reference voltage of the R2R DAC
SUP_ANA_RTUNE_CTRL.RT_DAC_MODE	W	0	Set the DAC mode for ADC function
SUP_ANA_RTUNE_CTRL.RT_DAC_CHOP	W	0	Set the RT DAC for ADC function

Note Many of these registers are a single bit in a 16-bit register. Therefore, before writing to a single bit, the whole 16 bits need to be read first and then written back, with only that one bit adjusted.

You can perform these register writes in any order and group them into a three-register write to reduce the number of writes: SUP_ANA_BG, SUP_ANA_RTUNE_CTRL, and SUP_DIG_RTUNE_DEBUG.

These register writes conflict with regular RTUNE operation and, therefore, must not be performed when the part is performing termination calibration.

1.2.2 Register Writes to Trigger the ADC Function

Table 1-2 describes the registers required to trigger the ADC function.

Table 1-2 Register Writes to Trigger the ADC Function

Order	Register Name	Read/Write	Value to be written	Comments
1	SUP_DIG_RTUNE_DEBUG.MAN_TUNE	W	1	Enable manual tuning
2	SUP_DIG_RTUNE_STAT.STAT[9:0]	R	N/A	Read the ADC output
3	SUP_DIG_RTUNE_DEBUG.MAN_TUNE	W	0	Disable manual tuning

Note Due to chopping of the bandgap circuit, the ADC read out could vary from read to read even at a fixed temperature. It is strongly recommended to perform multiple reads with the ADC function and take an average of the code for any given temperature measurement. A minimum of 64 reads is recommended.

Averaging of the code can be implemented with adder for accumulation and shifter for division. Therefore, we recommend the number of reads to be a power of 2.

These register writes conflict with regular RTUNE operation and, therefore, must not be performed when the part is performing termination calibration.

1.2.3 Register Writes to Disable Temperature Sensor

Table 1-3 describes the registers required to disable the Temperature Sensor.

Table 1-3 Register Writes to Disable the Temperature Sensor

Register Name	Read/ Write	Value to be written	Comments
SUP_ANA_BG.TEMP_SENSOR_EN	W	0	Disable temperature sensor circuit (by default, the temperature sensor is on)
SUP_ANA_RTUNE_CTRL.TEMP_SENSOR_SEL	W	0	Select the ATB voltage instead of Vctat for ADC measurement

Register Name	Read/ Write	Value to be written	Comments
SUP_ANA_RTUNE_CTRL.VREF_SEL	W	0	Select the VPH instead of Vref_bg as reference voltage of the R2R DAC
SUP_ANA_RTUNE_CTRL.RT_EN_FRCON	W	0	Remove force on RTUNE

Note

Many of these registers are a single bit in a 16-bit register. Therefore, before writing to a single bit, the whole 16 bits need to be read first and then written back, with only that one bit adjusted.

You can perform these register writes in any order and group them into a three-register write to reduce the number of writes: SUP_ANA_BG, SUP_ANA_RTUNE_CTRL, and SUP_DIG_RTUNE_DEBUG.

These register writes conflict with regular RTUNE operation and, therefore, must not be performed when the part is performing termination calibration.

2

Procedure and Measurements

This chapter includes the following sections:

2.1 Temperature Sensor Procedure

Using the registers described in Chapter 1, "Implementation", follow steps 1 and 2 presented below to calculate the temperature from the code.

1. Calibrate the Reference Look-Up Table (reference plot as shown in "Temperature Sensor Measurement Example" on page 14):
   a. Enable the temperature sensor (see Table 1-1 on page 10) at a known Temperature.
   b. Wait 50us.
   c. Trigger the ADC function and read the output (see Table 1-2 on page 11 for registers).
   d. Accumulate ADC codes from the previous step.
   e. Repeat b and c a total of 64 times.
   f. Disable the temperature sensor (see Table 1-3 on page 11 for registers).
   g. Divide the accumulated code by 64 to get the average code.
   h. Repeat a through g for all temperatures.
2. Perform temperature sensing in mission mode, as shown in step 1:
   a. Repeat a through g, as indicated in step 1.
   b. Use the slope in the reference plot obtained from step 1 to estimate the temperature from the code obtained in step 2a.

2.2 Temperature Sensor Measurement Example

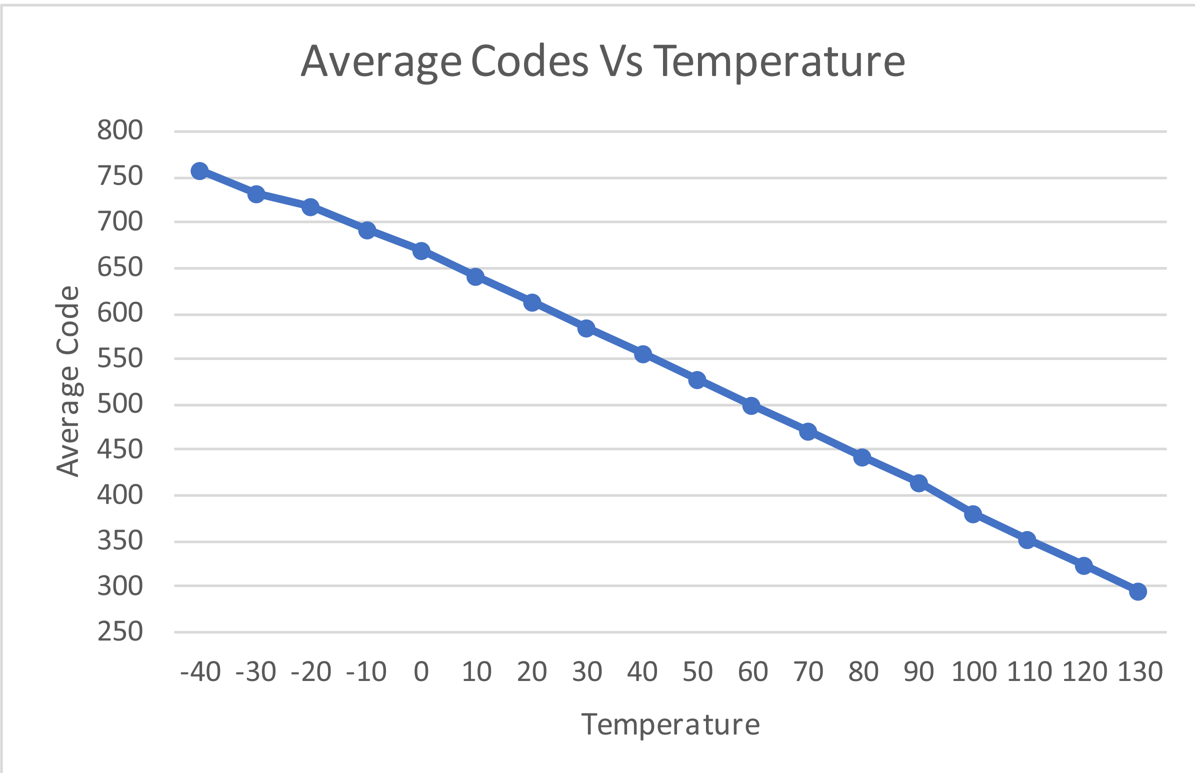
This section presents a sample temperature sensor reference look-up table for your reference (measurements are based on the Synopsys test chip). The following example is for illustration only. You must create the code versus temperature calibration table for your product.

Table 2-1 Temperature Versus Average Code Reference Look-Up Table

Temperature, ^{o}C	Average Code
-40	759
-30	733
-20	718
-10	694
0	670
10	641
20	614
30	585
40	557
50	528
60	500

Temperature, ^{o}C	Average Code
70	471
80	442
90	413
100	381
110	353
120	323
130	294

Figure 2-1 Temperature Versus Average Code Reference Plot



Note Depending on ASIC self-heating, there may be a difference between the silicon junction temperature and the package lead temperature.