Audio Clock Recovery Unit v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG335 (v1.0) May 22, 2019

XILINX®

Table of Contents

Chapter 1

Introduction

The Xilinx® LogiCORE™ Audio Clock Recovery Unit is a soft IP core for use with the Vivado® Design Suite. It provides an easy mechanism to recover the audio sampling clock from a given reference clock. It can be used with HDMI or DisplayPort receivers to recover the audio sampling clock.

Features

- A sampling clock recovery from a reference clock
- Fixed audio sampling clock recovery
- Loop control based audio sampling clock recovery
- Compatibility with HDMI and DisplayPort

IP Facts

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family[1]	UltraScale+™, UltraScale™, Zynq®-7000 SoC, Zynq® UltraScale+™ MPSoC, and 7 series FPGAs
Supported User Interfaces	AXI4-Lite
Resources	Performance and Resource Use web page
Provided with Core	
Design Files	Encrypted RTL
Example Design	System Verilog
Test Bench	System Verilog
Constraints File	XDC file delivered with IP generation
Simulation Model	Encrypted RTL
Supported S/W Driver	N/A
Tested Design Flows[2]	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Not Provided
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado® IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

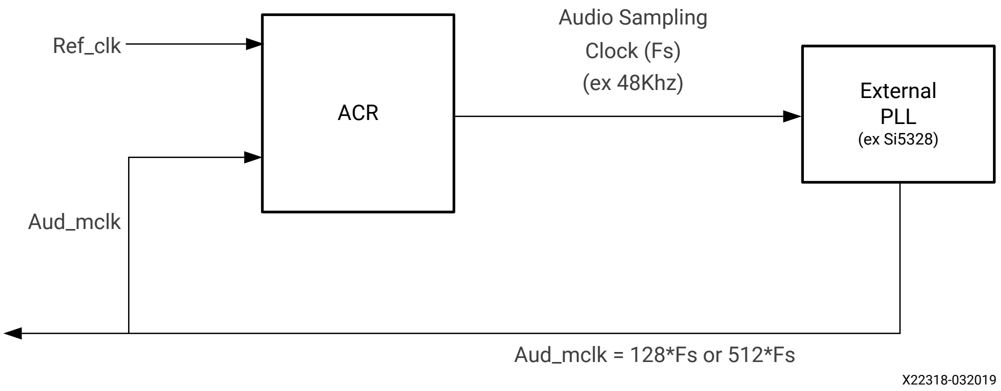
Overview

The Audio Clock Recovery Unit (ACR) core provides an easy mechanism to recover the audio sampling clock from a video reference clock. This is useful in recovering the audio sampling clock at the receiver end in HDMI and DisplayPort protocols.

Applications

The Audio Clock Recovery Unit core can be used in HDMI and DisplayPort systems to implement complete audio solution. The recovered audio sampling clock can be used to generate a clock to drive I2S or SPDIF. The application of the ACR unit is shown in the following figure:

Figure 1: **Application of the ACR Unit**



Licensing and Ordering

This Xilinx® LogiCORE™ IP module is provided at no additional cost with the Xilinx Vivado® Design Suite under the terms of the Xilinx End User License.

Note: To verify that you need a license, check the License column of the IP Catalog. Included means that a license is included with the Vivado® Design Suite; Purchase means that you have to purchase a license to use the core.

For more information about this core, visit the Audio Clock Recovery Unit product web page.

Information about other Xilinx® LogiCORE™ IP modules is available at the Xilinx Intellectual Property page. For information about pricing and availability of other Xilinx® LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Performance and Resource Use

For full details about performance and resource use, visit the Performance and Resource Use web page.

Port Description

Port Names

Table 1: **Port Names and Description**

Port Name	I/O	Clock	Description
s_axi_ctrl_aclk	I	Clock	Input clock for AXI4-Lite interface
s_axi_ctrl_aresetn	I	Reset	active-Low reset for AXI4-Lite interface
s_axi_ctrl*		s_axi_ctrl	AXI4-Lite Interface
acr_clk	I	Clock	Input for ACR data
acr_resetn	I	Reset	active-Low reset input to reset the ACR logic
acr_cts	I	CTS/Naud	Input for CTS data (HDMI) or NAUD (DisplayPort)
acr_n	I	N/Maud	Input for N data (HDMI) or MAUD (DisplayPort)
acr_valid	I	ACR valid	Single bit data indicating presence of valid N and CTS value
aud_mclk	I	Clock	Audio clock generated from the external clock chip
aud_mrst	I	Reset	active-High reset input to reset the Mclk logic
ref_clk	I	Clock	Reference input clock from which the audio sampling clock is to be recovered
ref_clk_resetn	I	Reset	active-Low reset input to reset the ref clk logic
fifo_datacount_in	I	FIFO count	This is the data count of the FIFO that is used to buffer the audio data. This has to be connected to a valid signal when using ACR in loop control mode
aud_clk_out	O	-	This is the recovered audio sampling clock
irq	O	Interrupt output	This is currently unused

Register Space

Table 2: **Register Address Space**

Address (hex)	Register
0x00	Core Version
0x04	Core Configuration
0x08	Core Control
0x20	Mode Register
0x30	FIFO Setpoint
0x34	Deltasum Limit
0x38	Delta Size
0x3C	Sample Control
0x40	FIFO Avg Control
0x50	N USer
0x54	CTS User
0x70	OUT DIV

Core Version Register (0x00)

This register is reserved for future use.

Table 3: **Core Version Register (0x00)**

Bit	Default Value	Access Type	Description
31:0	0	RO	Reserved

Core Configuration Register (0x04)

This register is reserved for future use.

Table 4: **Core Configuration Register (0x04)**

Bit	Default Value	Access Type	Description
31:0	0	RO	Reserved

Core Control Register (0x08)

This register provides capability to enable/disable the core operation.

Table 5: **Core Control Register (0x08)**

Bit	Default Value	Access Type	Description
31:1	0	RO	Reserved
0	0	R/W	Writing 0x1 enables the core operation.

Mode Register (0x20)

This registers allows you to program the mode of the clock recovery. You can either choose to use the ACR details from the ports or registers.

Table 6: **Mode Register (0x20)**

Bit	Default Value	Access Type	Description
31:3	0	RO	Reserved
2:0	0	R/W	3'b000 - External Fixed mode. N/CTS (Maud/Naud) values taken from input ports 3'b001 - User Fixed Mode. N/CTS (Maud/Naud) values taken from registers 3'b100 - Extarnal Loop Control mode. N/CTS (Maud/Naud) values taken from input ports 3'b101 - User Loop Control mode. N/CTS (Maud/Naud) values taken from registers.

FIFO Setpoint Register (0x30)

This register is used to specify the FIFO setpoint in case of a loop control mode. This value specifies the threshold of the audio FIFO buffer that should always be maintained. In loop control mode, the ACR tries to adjust the output to maintain the FIFO occupancy at the current level.

Table 7: **FIFO Setpoint Register (0x30)**

Bit	Default Value	Access Type	Description
31:16	0	RO	Reserved
15:0	0	R/W	Specify the FIFO setpoint value. This value should be equal to the threshold value of the audio buffer. This register is used only in loop control mode.

DeltaSum Limit Register (0x34)

This register allows the you to specify the maximum limit of deltasum. In loop control mode, the ACR internally adjusts the value of N or Maud to vary the output frequency. This register specifies the maximum limit of this variation. For example, if N=6000 and this register is programmed as 200, then the ACR internally varies the value of N to a maximum value of 6200 and a minimum value of 5800.

Table 8: **DeltaSum Limit Register (0x34)**

Bit	Default Value	Access Type	Description
31:16	0	RO	Reserved
15:0	0	R/W	Specify the maximum limit of the sum delta. This is the max amount of variation required in N or Maud. This is used only in loop control mode.

DeltaSize Register (0x38)

This register allows you to specify the minimum amount by which the value of N or Maud is incremented/decremented internally. For example, if this value is 10 and N=6000, then N would be incremented/decremented by a value of 10 until the DeltaSum limit is hit. In other words, this is the granularity of the variation of N.

Table 9: **DeltaSize Register (0x38)**

Bit	Default Value	Access Type	Description
31:16	0	RO	Reserved
15:0	0	R/W	Specify the value of Delta Size.

Sample Control Register (0x3C)

This is the time interval at which ACR takes a decision to either increment or decrement the N or Maud in loop control mode. This counter runs on aud_mclk.

Table 10: **Sample Control Register (0x3C)**

Bit	Default Value	Access Type	Description
31:16	0	RO	Reserved
15:0	0	R/W	Specify the value of the sample interval.

FIFO Average Control Register (0x40)

This register lets you specify the number of clock cycles to be used for averaging the FIFO count. At the end of this, if the average is more than the FIFO setpoint, then ACR would vary the N or Maud to increase the output. If the average is less than the FIFO setpoint, then ACR would vary the N or Maud to decrease the output frequency.

Table 11: **FIFO Average Control Register (0x40)**

Bit	Default Value	Access Type	Description
31:4	0	RO	Reserved

Table 11: **FIFO Average Control Register (0x40)** *(cont'd)*

Bit	Default Value	Access Type	Description
3:0	0	R/W	Specify the number of clock cycles to be used for averaging. Num of clock cycles of equal of to power of 2 of this value.

N/Maud USER Register (0x50)

This register allows you to specify the N or Maud value when it is not available as input.

Table 12: **N/Maud USER Register (0x50)**

Bit	Default Value	Access Type	Description
31:16	0	RO	Reserved
15:0	0	R/W	Specify the N or Maud value. This register is application only when the mode is set to User.

CTS/Naud Register (0x54)

This register allows you to specify the CTS/Naud value when the same is not available on input ports.

Table 13: **CTS/Naud Register (0x54)**

Bit	Default Value	Access Type	Description
31:16	0	RO	Reserved
15:0	0	R/W	Specify the CTS or Naud value. This is is used only when the ACR is set in User mode.

OUT_DIV Register (0x70)

Program this register to specify the output divider. This value can be tweaked to get the desired output frequency.

Table 14: **OUT_DIV Register (0x70)**

Bit	Default Value	Access Type	Description
31:8	0	RO	Reserved
7:0	0	R/W	Specify the value of the final divider. OutputDivider = 2*OUT_DIV

Chapter 4

Designing with the Core

This section includes guidelines and additional information to facilitate designing with the core.

The `aud_clk_out` is the audio sampling clock signal recovered from the reference clock. This signal is of very low frequency and is used to generate `aud_mclk`. Hence, an external PLL is always required to generate the `aud_mclk` in MHz range.

The following equation explains how an `aud_clk_out` is generated for HDMI:

Aud_clk_out = (Fref * N/(2*CTS)) * (1/OutputDivider)

Fref = RX TMDS Clock frequency

The following equation explains how an `aud_clk_out` is generated for DisplayPort:

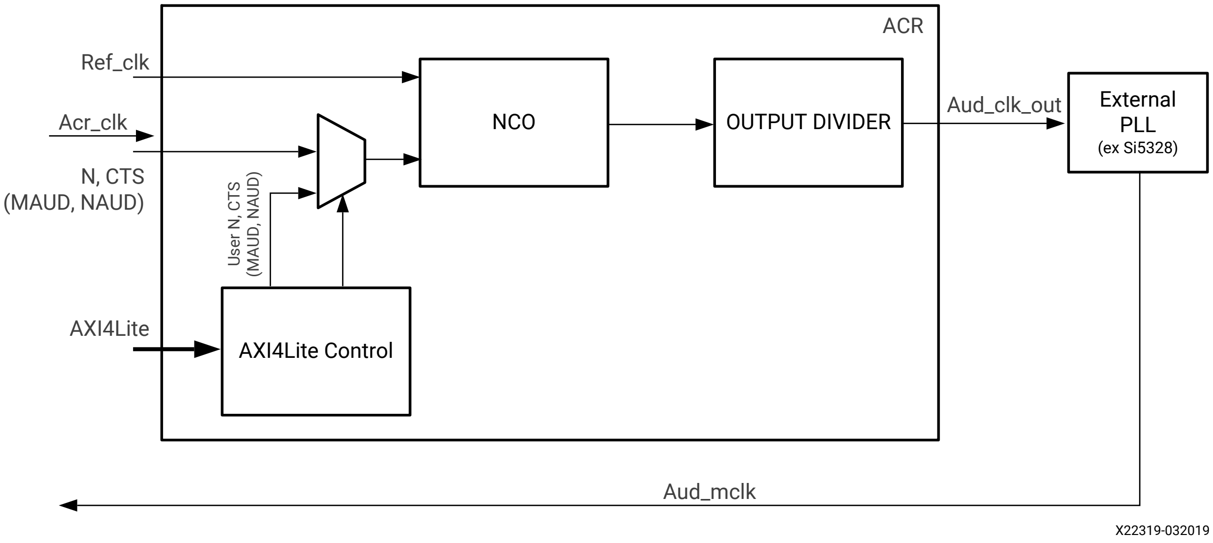
Aud_clk_out = (Fref * Maud/(2*Naud)) * (1/OutputDivider)

Fref = rxoutclk frequency

General Design Guidelines

The Audio Clock Recovery Unit IP can be used in two modes. The fixed mode can be used when the reference clock from the source is available at the receiver end. For HDMI solution, the receiver gets the same TMDS clock as source. As such, this mode can be used for HDMI solutions. In the fixed audio clock recovery mode, the audio sampling clock is recovered based on the values of N and CTS. This is good for HDMI as the TMDS clock is available at the receiver. The following figure shows the typical usage for HDMI:

Figure 2: **Audio Clock Recovery - Fixed Mode**

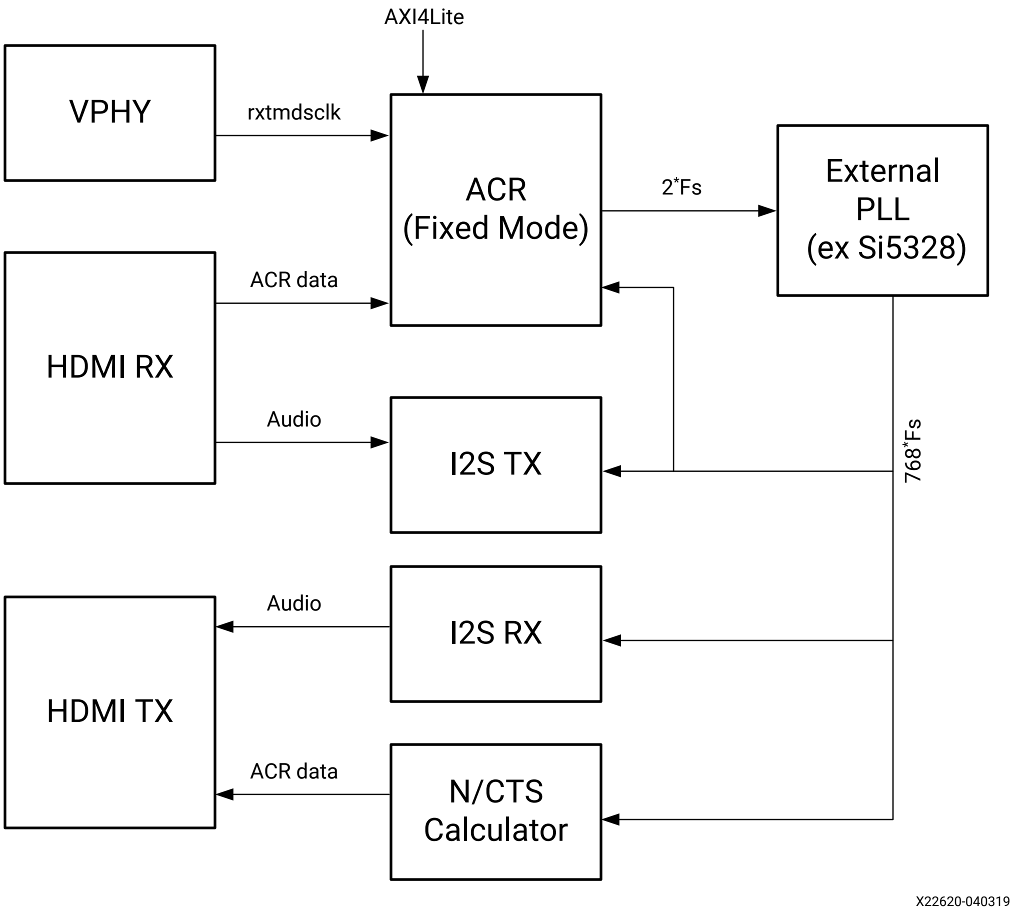


Following is the typical programming sequence in the fixed mode:

1. Set the mode register.
2. Program the registers 0x50 and 0x54 if operating in the user mode.
3. Program the register 0x70 to specify the divider value.
4. Start the ACR unit by writing 0x10 to 0x08 registers.

The following figure shows the clock connectivity in the HDMI example design:

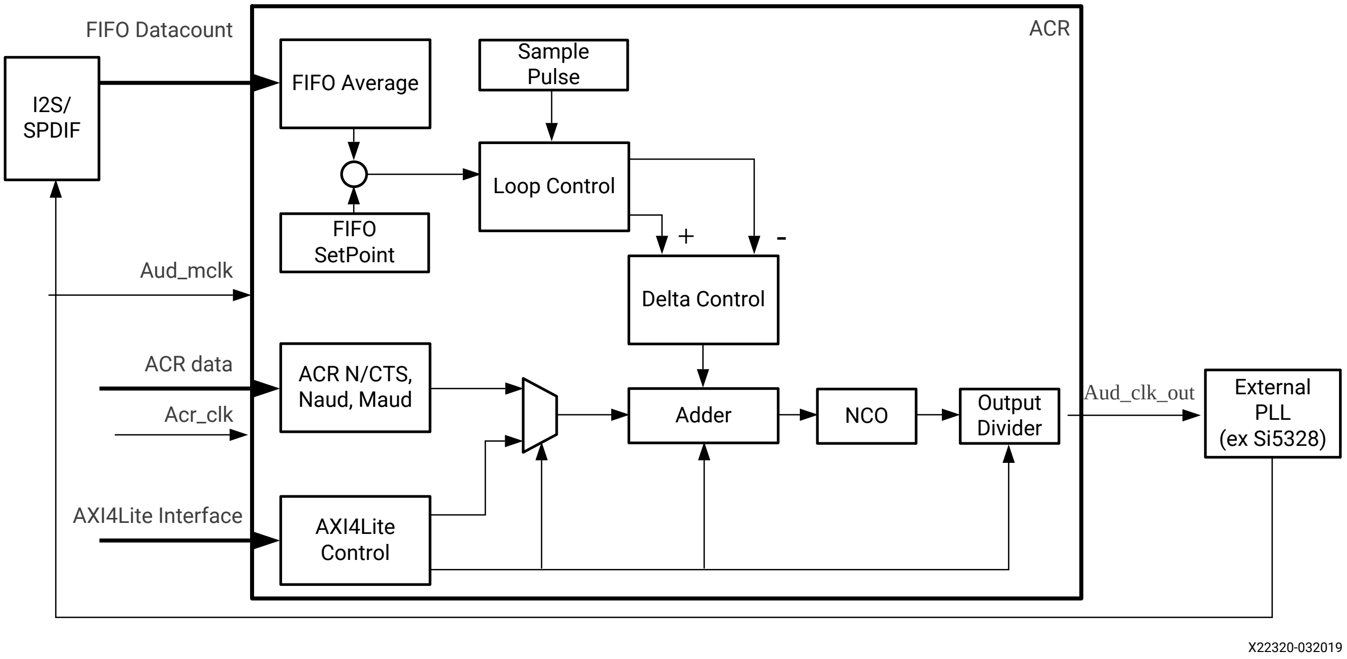
Figure 3: **Clock Connectivity in the HDMI Example Design**



In the case of a DisplayPort, the receiver operates on a clock that is completely independent from the clock in the transmitter. During the audio clock recovery, you should ensure to match the jitter/ppm variation. This is because the Maud and Naud values are calculated based on the clocks in the transmitter. There is no guarantee that the clocks at the receiver will have similar properties. The loop control mode of the ACR is ideal for such a scenario. In this mode, the ACR unit continuously adjusts to match the incoming rate thereby ensuring a reliable audio clock recovery that avoids any overrun or under run of the audio buffer.

In the loop control based audio clock recovery mode, the ACR unit is adjusted based on the FIFO occupancy level. This is helpful in adjusting the jitter/PPM when the reference clock at the receiver end is independent of the transmitter. This is suited for the DisplayPort. In this mode, the ACR output varies based on the FIFO occupancy level, which in turn affects the output of an external chip thereby adjusting the jitter/PPM. For example, for a 48 KHz audio, the output clock could vary continuously between 47 KHz and 49 KHz in order to maintain the FIFO level.

Figure 4: **Audio Clock Recovery - Loop Control Mode**



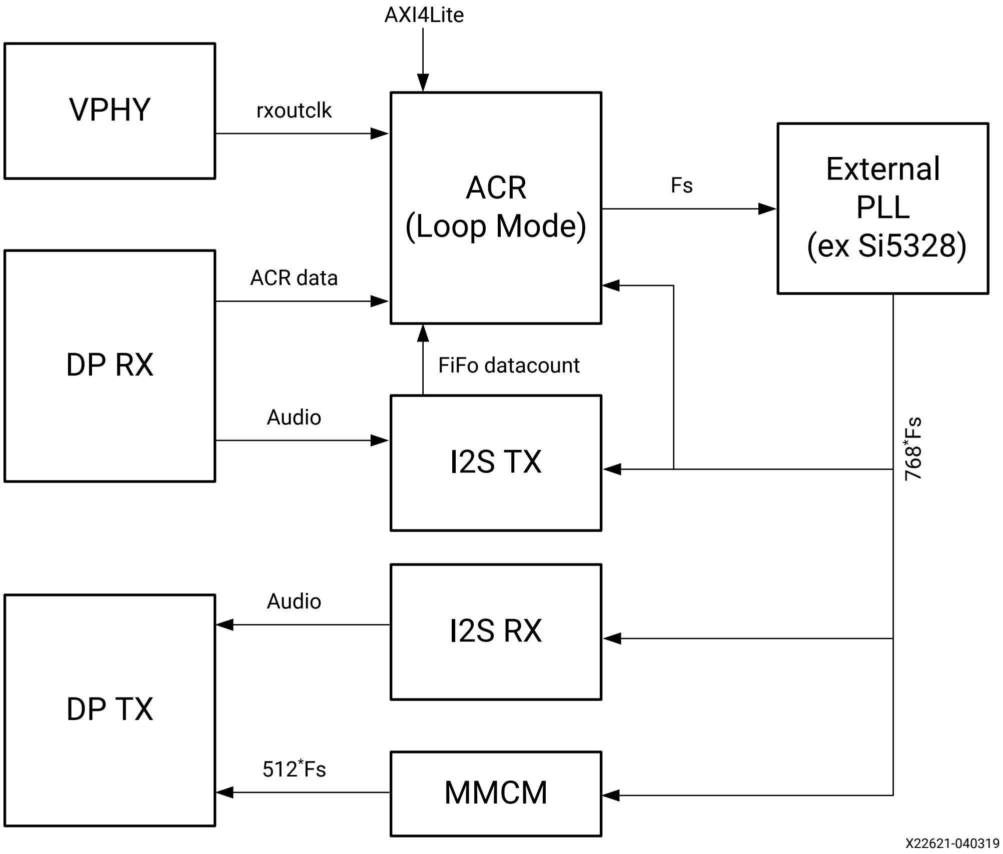
Following is the typical programming sequence in the loop control mode:

1. Set the mode register.
2. Program the registers 0x50 and 0x54 if operating in the user mode.
3. Program the register 0x70 to specify the divider value.
4. Program the FIFO set point register 0x30.
5. Program the registers 0x34 and 0x38 to specify the delta limit and size.
6. Program the registers 0x3C and 0x40 to specify the sample pulse control and FIFO average control.
7. Start the ACR unit by writing 0x1 to 0x08 registers.

***Note*:** In both the operating modes, an external clock chip is needed. Programming of the clock chip is not covered in this product guide.

The following figure shows the clock connectivity in the DisplayPort example design:

Figure 5: **Clock Connectivity in the DisplayPort Example Design**



Clocking

Table 16: **Clocks and Description**

Clock	Description
s_axi_ctrl_aclk	This is the AXI4-Lite interface clock. This can be of any frequency. Typically in a system, this clock is shared with other peripherals.
ref_clk	This is the source clock for audio sampling clock recovery. This is the output of the vid_phy_controller. In case of HDMI it is the TMDS clock (`rx_tmds_clk`). In case of DisplayPort, it is the link clock (`rxoutclk`).
acr_clk	This is the clock that is used to drive the ACR data from HDMI or DisplayPort.
aud_mclk	This is the clock generated by the external clock chip. The frequency of this clock is decided based on the audio peripherals such as DAC/ADC and/or I2S. This clock is typically an integer multiple of 128*Fs.
aud_clk_out	This is a reference clock signal to be connect to an external clock chip. This signal is in KHz range and should not be used inside an FPGA.

Notes:

1. For more details on clocking and usage, please refer HDMI and DisplayPort example designs in Chapter 4: Designing with the Core.

Resets

The s_axi_ctrl_aresetn is the AXI4-Lite active-Low reset. It resets all the registers to their default values. The ref_clk_resetn and acr_resetn are active-Low resets that reset the ref_clk and acr_clk logic respectively. The aud_mrst is a active-High reset that resets the loop control logic.

Design Flow Steps

This section describes customizing and generating the core, constraining the core, and the simulation, synthesis, and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
- *Vivado Design Suite User Guide: Designing with IP* (UG896)
- *Vivado Design Suite User Guide: Getting Started* (UG910)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado® Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) and the *Vivado Design Suite User Guide: Getting Started* (UG910).

Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896).

Constraining the Core

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Appendix A

Debugging

This appendix includes details about resources available on the Xilinx® Support website and debugging tools.

If the IP requires a license key, the key must be verified. The Vivado® design tools have several license checkpoints for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with an error. License checkpoints are enforced by the following tools:

- Vivado Synthesis
- Vivado Implementation
- write_bitstream (Tcl command)

Note: IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.

Finding Help on Xilinx.com

To help in the design and debug process when using the core, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support. The Xilinx Community Forums are also available where members can learn, participate, share, and ask questions about Xilinx solutions.

Documentation

This product guide is the main document associated with the core. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx® Documentation Navigator. Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use keywords such as:

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the Audio Clock Recovery Unit Core

AR 54488.

Technical Support

Xilinx provides technical support on the Xilinx Community Forums for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To ask questions, navigate to the Xilinx Community Forums.

Debug Tools

There are many tools available to address Audio Clock Recovery Unit design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx® devices.

The Vivado logic analyzer is used to interact with the logic debug LogiCORE IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908).

Appendix B

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

***Note*:** For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this guide:

1. *DisplayPort 1.4 RX Subsystem Product Guide* (PG300)
2. *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235)
3. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
4. *Vivado Design Suite User Guide: Designing with IP* (UG896)
5. *Vivado Design Suite User Guide: Getting Started* (UG910)
6. *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
05/22/2019 Version 1.0	
Initial Xilinx release.	N/A

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https://

www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright