



Lab4 Intro

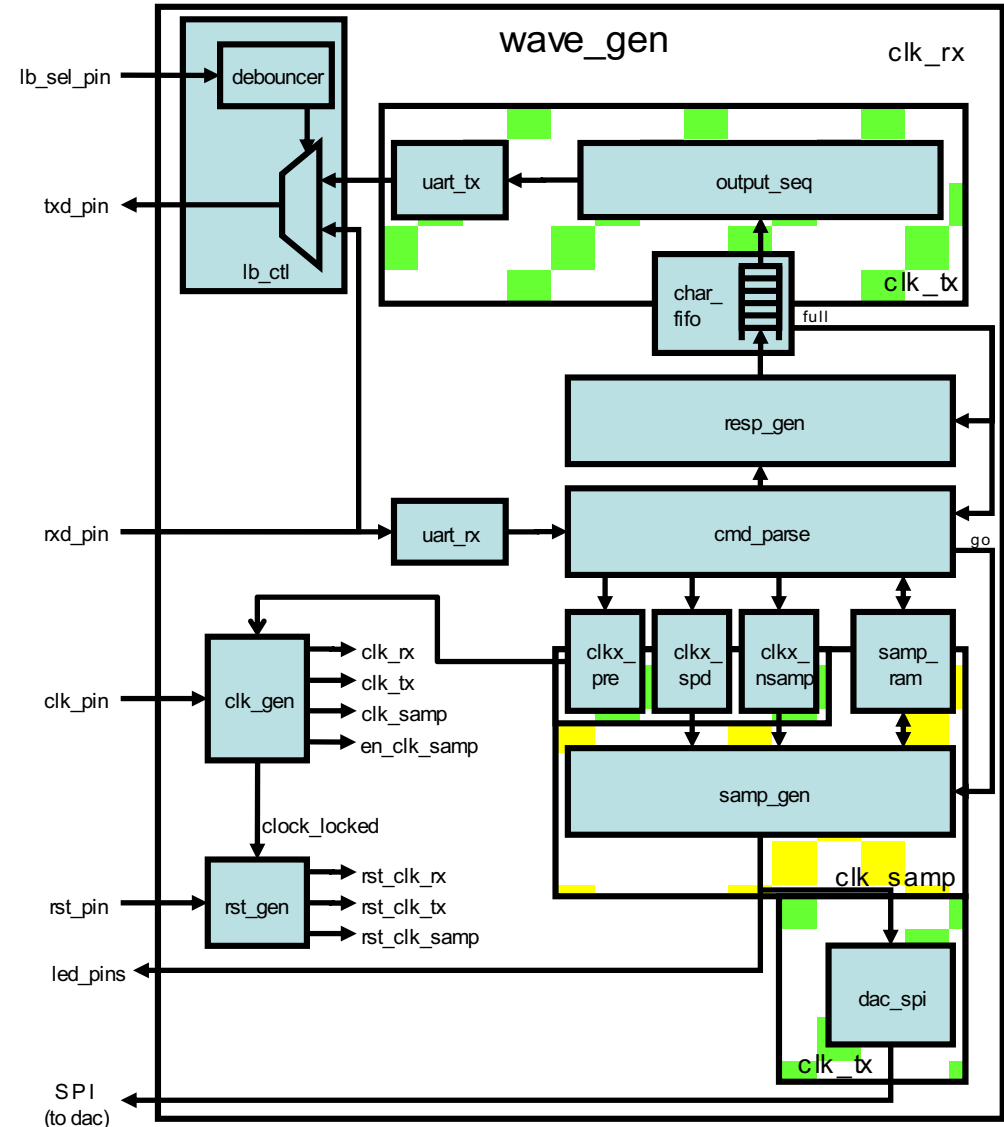
Using IP Catalog

Introduction

- ▶ In this lab you will use the IP Catalog to generate a clock resource. You will instantiate the generated core in the provided waveform generator design

The Design

- ▶ The waveform generator in this design is intended to be a “standalone” device that is controlled via a PC (or other terminal device) using RS-232 serial communication. The design described here implements the RS-232 communication channel, the waveform generator and connection to the external DAC, and a simple parser to implement a small number of “commands” to control the waveform generation



The Design Functionality

- ▶ This design "records" specific information via RS-232 serial communication and stores this data in an on-chip memory
- ▶ After data has been stored, it can be retrieved via the RS-232 communications channel, or "played" out via a bank of LEDs or a DAC
- ▶ Receives RS-232 serial data at 115200 baud (no parity, 8 data bits, no handshaking)
- ▶ Handled in *uart_rx*, which is simple state machine and an over sampler
- ▶ Small command set controls how information is stored and played back

The Design

- ▶ The *wave_gen* design consists of several top level blocks:
 - Clock and reset management
 - UART receiver and transmitter
 - Command parsing
 - Response generation
 - Character FIFO
 - Sample RAM and generator
 - SPI generator
 - Various clock crossing modules

The Clock Domains

- ▶ The *wave_gen* design uses three clock domains
 - Receive clock (*clk_rx*)
 - Transmit clock (*clk_tx*)
 - Sample clock (*clk_samp*)
- ▶ All clocks are derived from a single clock input pin (*clk_pin*) using a single MMCM
 - The clock input depends on the board
- ▶ The receive clock runs at the input clock frequency (100 MHz)
 - The receive clock is assumed to be asynchronous to the transmit and sample clocks
- ▶ The transmit clock runs either at the same frequency or at 31/32 of that frequency
- ▶ The sample clock is a decimated version of the transmit clock
 - Rate is determined by the *prescale* value

The wave_gen Commands

Cmd	Input	Response	Description
*W	aaaavvvv	-OK or -ERR	03ff ≥ aaaa ≥ 0000. Value “vvvv” is written into RAM at location “aaaa” and “-OK” is return.
*R	aaaa	-hhhh dddd or -ERR	03ff ≥ aaaa ≥ 0000. If in range, then the value at “aaaa” is returned in hex and decimal.
*N	vvvv	-OK or -ERR	0400 ≥ vvvv ≥ 0001. Specifies the number of samples before recycling.
*P	vvvv	-OK or -ERR	ffff ≥ vvvv ≥ 0020. Specifies prescaling value to divide <i>clk_tx</i> by to produce <i>clk_samp</i> .
*S	vvvv	-OK or -ERR	ffff ≥ vvvv ≥ 0001. Specifies “speed” value to divide <i>clk_samp</i> by to produce the rate of read from RAM.
*n/*p/*s		-hhhh dddd	Returns current value of nsamp, prescale, and speed.
*G		-OK	Triggers a single pass through nsamp memory locations.
*C		-OK	Starts continuous triggering.
*H		-OK	Halts continuous loop at end of current cycle.

Procedure

- ▶ Create the project
- ▶ Generate and instantiate a clock generator module
- ▶ Implement the design
- ▶ Configure the target board and verify the functionality

Summary

- ▶ In this lab, you learned how to add an existing IP during the project creation. You also learned how to use IP Catalog and generate a core. You then instantiated the core in the design, implemented the design, and verified the design in hardware. You also used the IP Integrator capability of the tool to generate a FIFO and then use it in the HDL design.



Thank You

Disclaimer and Attribution

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

© Copyright 2022 Advanced Micro Devices, Inc. All rights reserved. Xilinx, the Xilinx logo, AMD, the AMD Arrow logo, Alveo, Artix, Kintex, Kria, Spartan, Versal, Vitis, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

