Digital Integrated Circuits

Lab 3: Synthesis and Formality

Objective

Learn how to synthesize a digital design.

Introduction

Laboratory tasks

1. Download lab3.zip file and unpack it to your working directory.

The following files list shall be available after:

```
├── filelist
│   ├── simulation.flist
│   └── vcs_compile.sh
├── formality
│   └── fv.tcl
├── include
│   └── serializer_cc_constants.sv
├── Makefile
├── rtl
│   ├── serializer_clkdiv.sv
│   ├── serializer_dig.sv
│   ├── serializer_prbs.sv
│   ├── serializer.sv
│   ├── serializer_sync_clkin_to_clkvco_div2.sv
│   ├── serializer_sync_reset.sv
│   ├── serializer_sync_single_bit.sv
│   └── serializer_top.sv
├── spyglass
│   ├── clocks.sgdc
│   ├── list_rtl_files.txt
│   ├── local_setup_options.tcl
│   ├── manual.sgdc
│   ├── ports.sgdc
│   ├── resets.sgdc
│   ├── run_cdc.sh
│   ├── sg.prj
│   ├── spyglass_engineering_council_rules.tcl
│   ├── spyglass_engineering_council_sg_rules.tcl
│   ├── spyglass_msip_coding_guidelines_rules.tcl
│   ├── spyglass_msip_council_rules.tcl
│   └── waivers.tcl
├── synthesis
│   └── dc.tcl
├── testbench
│   └── dig_functional_test.sv
└── work
```


2. **Source Synopsys tools**

3. **Run % make help to understand how the makefile works and what it does.**

   - **Make sure you are at LAB3 directory, where the make is located**

4. **Open the file dc.tcl inside the folder synthesis this is the main script where you will be making changes.**
5. **The first step should be to update in to update the DESIGN_REF_PATH variable to ./edk.**
6. **After this you should check the constraints powerpoint to be able to complete the constraints where the TODO is located.**
7. **You should check the deign for timing violations this report is located inside the following directory ./synthesis/results/reports**

Formality:

1. When you finish the synthesis you should follow up with formal verification where you check if the gate level netlist is functionally the same as the RTL.
2. You should start with opening the script under the following dir /formality/fv.tcl
3. The first step should be to update in to update the DESIGN_REF_PATH variable to ./edk.
8. **Now just run the make run_formality**

1. High Speed Serializer

1.1. Functional overview

The circuit to implement is a synchronous 10-to-1 parallel to serial converter operating at a serial throughput of 1Gbps. Its digital interface receives a 10 bit bus and clock and outputs a differential analog serial signal to be connected to a PCB trace. Both the parallel clock and the data come from a generic video controller are synchronous with each other, while the high speed serial clock comes from a local PLL, operating at 2GHz, and has no timing relation with the input parallel clock (known frequency relation but unknown phase relation). To allow flexibility on the routing of the PCB traces, the positive and negative differential outputs can be swapped through a configuration pin. There's an additional test mode used for characterization purposes that allows the digital block to generate, automatically and independently, a Pseudo-Random Binary Sequence of modulo-7 (PRBS Mod7) of maximum length, described by the generator polynomial x^7+x^6+1.

There's no restriction about circuit latency and the partitioning is up to you as is the verification of the complete PHY (physical layer). You are required to use the top level pinout that is given to you, although the digital block's pinout is up to you to decide, as long as it complies with top level interface and internal digital-to-analog interface.

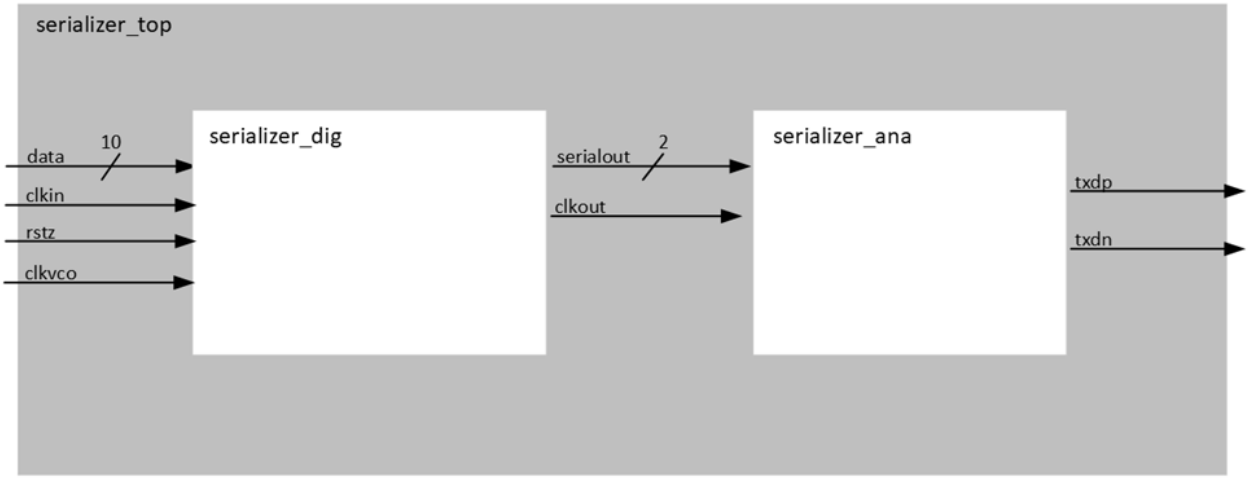

Figure 1 – Top level block diagram

2.2. serializer_top pin description

Signal Name	I/O	Bus Range	Description
VPH	I/O	-	Analog core voltage supply (1.0V)
GD	I/O	-	Analog core supply ground
VDD	I/O	-	Digital core voltage supply (1.0V)
VSS	I/O	-	Digital core supply ground
txdp	O	1	Analog non-inverting data output
txdn	O	1	Analog inverting data output
clkin	I	1	Input parallel clock.
rstz	I	1	Asynchronous reset. **Active low.**
data	I	10	Input parallel data. Synchronous with clkin.
clkvco	I	1	High speed serial clock.
invpolarity	I	1	Inverts txdp and txdn. **Active high.**
prbsenable	I	1	Enables internal PRBS generator. **Active high.**
scanin	I	1	Scan in. Used as serial data stream inputs, when macro is in Scan mode.

scanout	O	1	Scan out. Used as serial data stream outputs, when macro is in Scan mode.
scanclk	I	1	Scan clock. Clock source for scan mode.
scanrstz	I	1	Scan reset. Reset signal for scan mode. **Active Low**.
scanmode	I	1	Scan mode. Places the macro in scan mode when asserted. **Active high**.
scanen	I	1	Scan enable. Used to select scan operation mode. '1' – shift mode. '0' - capture mode.

Table 1 – serializer_top pin description

2.3. Timing diagrams

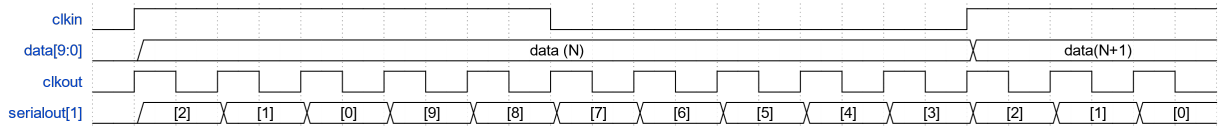

Figure 2 – Timing diagram of the serialization order

2.4. Physical constraints

The interaction between Controller and serializer_top is synchronous with clkin and respects the following timing interface:

- Minimum delay between the rising edge of clkin and the change of data[9:0] is: 0 ns
- Maximum delay between the rising edge of clkin and the change of data[9:0] is: 6 ns

- data[9:0] signals are stable after 6ns of the clkin rising edge and remain stable until the next clkin rising edge

Prbsenable and invpolarity input ports are related with test modes and, therefore, don’t have any relation with of the system clocks.