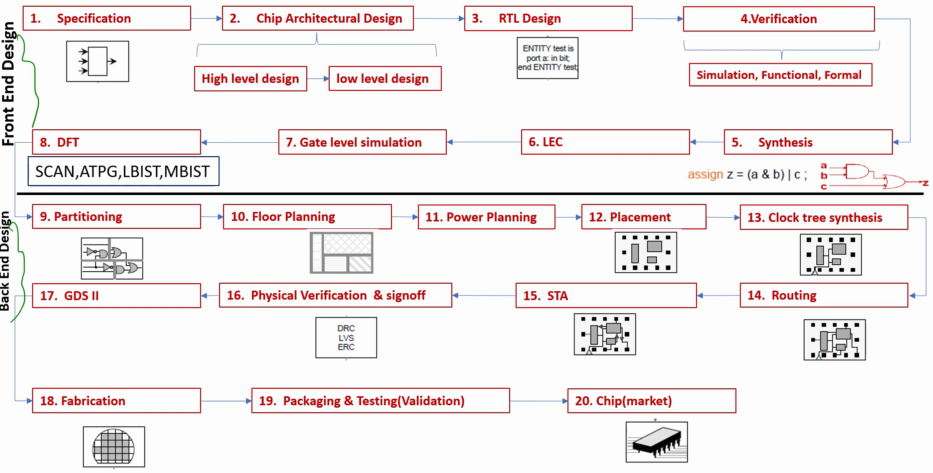
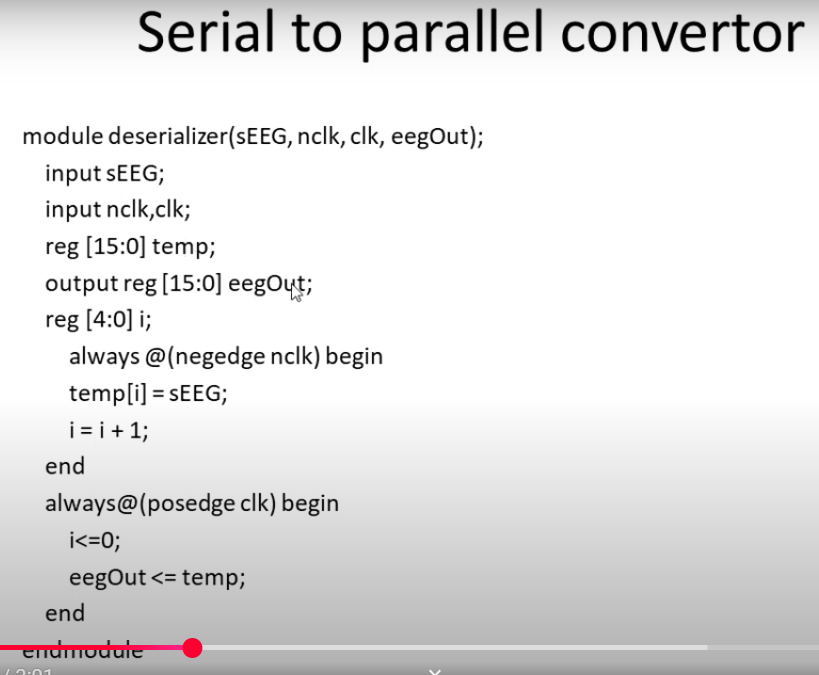
****

****

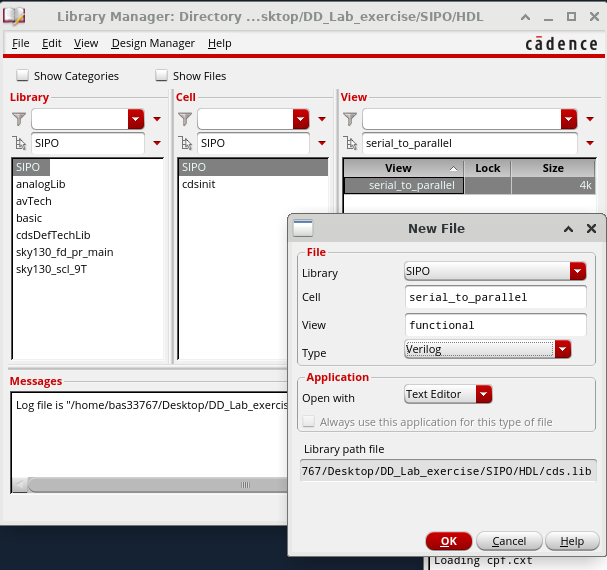
**Design Entry** (HDL 🡺 Verilog file)

**Plain Text File:**

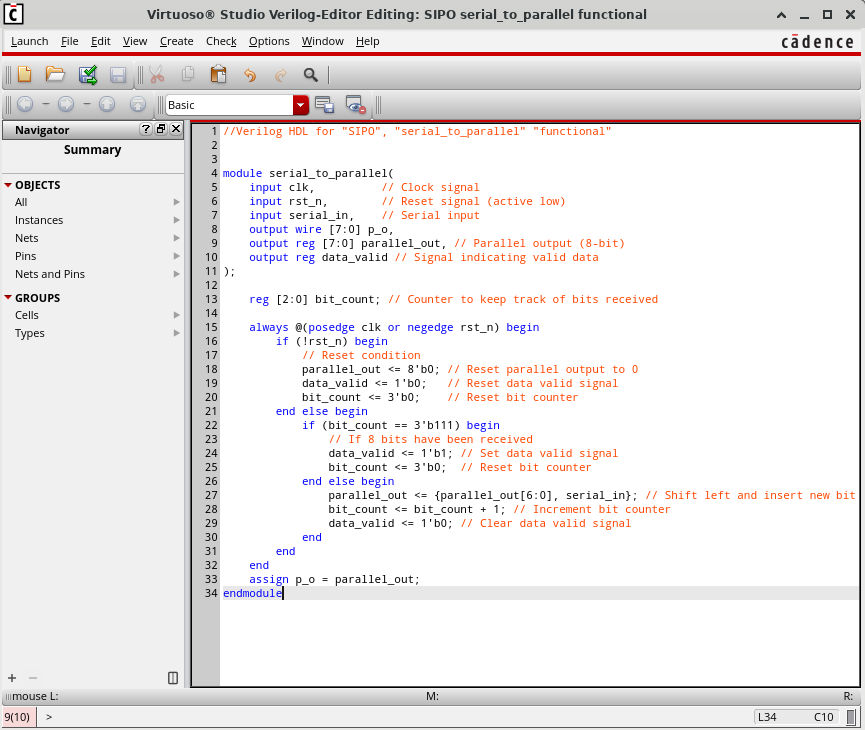
The Design entry in the form of HDL / Verilog file can be done in plain text file with “\*.v” extension. In that case compilation can be done invoking the design in XCELIUM.

In Virtuoso Studio:

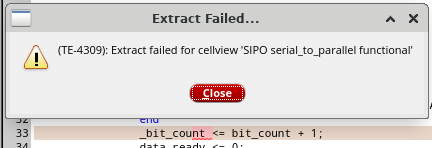
Like a typical analog design a new cell view can be created from the “File” menu. The Type should be “Verilog” and open with Text Editor.



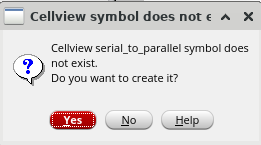
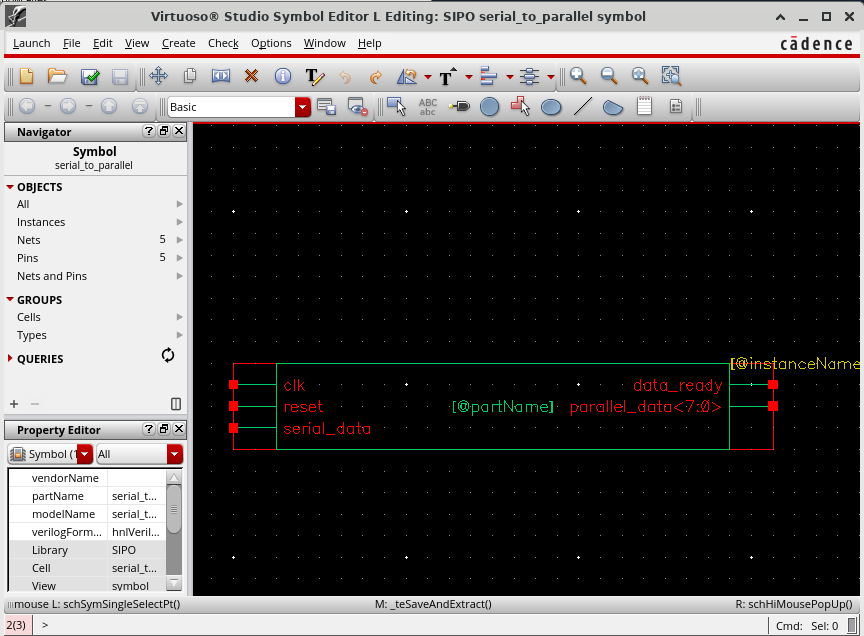
The “Text Editor” should be looked like the following—



The design can be build and checked with the build button (). If there is any syntax error it will be highlighted as follows—



If everything is good it will ask to create a symbol of the Design (only for new cellview-if not symbol already created).

 It could be useful for Mixed Signal design/Analysis which is out of scope of this exercise.

Similarly a testbench can be created. But it is not convenient to use testbench in Virtuoso.

Verification:

Now HDL/Verilog is available. Next step is to verify it. Verification can be formal or functional. Advance formal verification can be done with Cadence JasperGold which is not available. Basic verification can be done using XCELIUM.

Pre-Synthesis Verification:

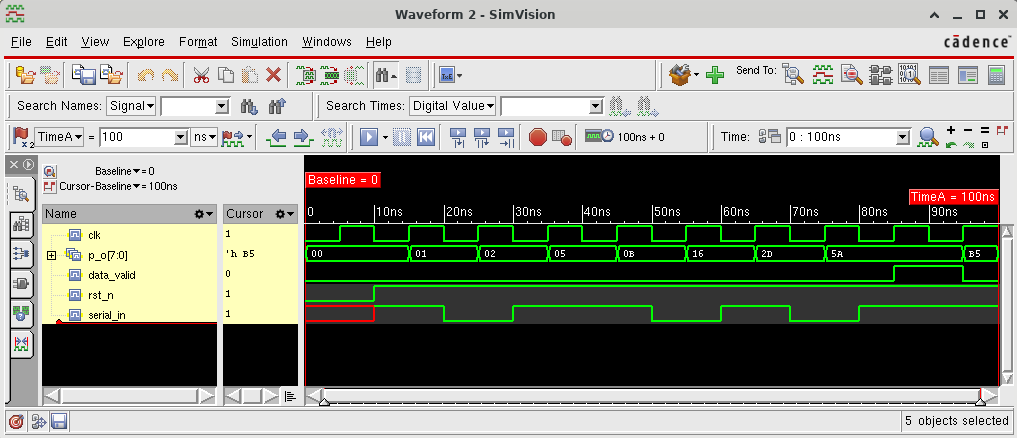
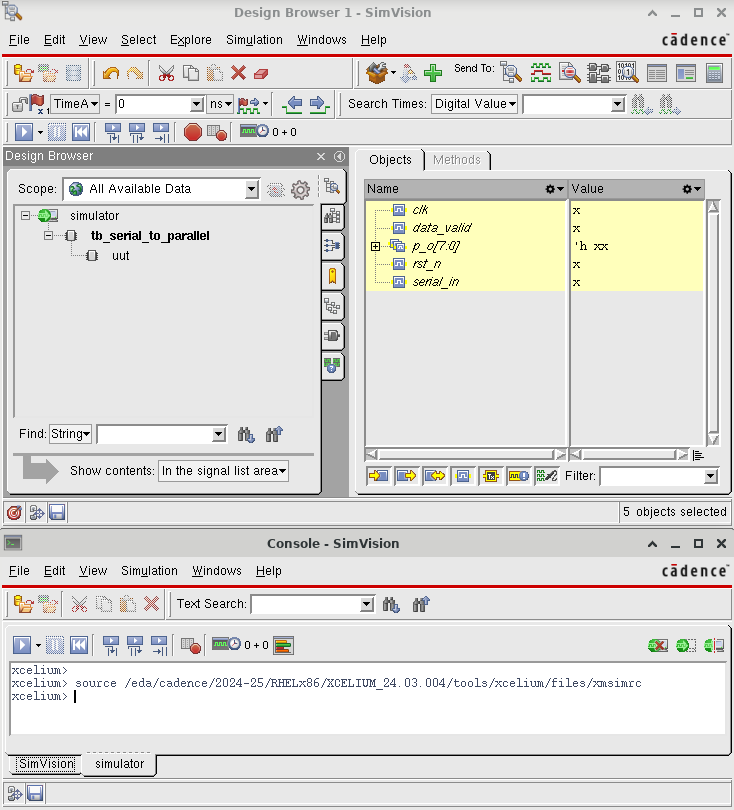
For verification XCELIUM will be used. Provided that the Verilog file and the testbench available the following command can invoke the SimVision control panels:

>> xrun SIPO\_tb.v -access +rwc -64bit -gui &

Here SIPO\_tb.v is the testbench file which instantiate and included the SIPO.v file. It also can be written in the following way if the SIPO.v file is not included in the testbench but instantiated:

>> xrun SIPO\_tb.v SIPO.v -access +rwc -64bit -gui &

The “-64bit” is optional. Could be necessary when the design is too big.



Synthesis:

<https://www.vlsi-expert.com/2011/02/synopsys-design-constraints-sdc-basics.html>

Für Synthesis, zuerst bracht man die SDC Datai und Liberty Datai von PDK. Eine grundlegende SDC Datai ist wie folgendes:

*## SDC Version  
set sdc\_version 2*

*# Define the clock with a period of 6.67 ps (150 GHz clock speed)  
create\_clock -period 6.67 -name clk -waveform {0.0 3.335} [get\_ports clk]*

*# Set the load capacitance for the output ports (assuming a load of 0.5 for demonstration)  
set\_load 0.5 [get\_ports p\_o]*

*# Set the maximum fanout for the output ports  
set\_max\_fanout 5 [get\_ports p\_o]*

*# Set the maximum transition time for the output ports  
set\_max\_transition 0.2 [get\_ports p\_o]*

*# Set the clock transition time  
set\_clock\_transition 0.03 [get\_clocks clk]*

*# Set the clock uncertainty for setup and hold  
set\_clock\_uncertainty -setup 0.06 [get\_clocks clk]  
set\_clock\_uncertainty -hold 0.03 [get\_clocks clk]*

*# Set the input delay for the input ports (assuming a maximum input delay of 1 ps)  
set\_input\_delay -max 1.0 -clock clk [get\_ports {serial\_in rst\_n}]*

*# Set the output delay for the output ports (assuming a maximum output delay of 2 ps and minimum of 1.5 ps)  
set\_output\_delay -max 2.0 -clock clk [get\_ports p\_o]  
set\_output\_delay -min 1.5 -clock clk [get\_ports p\_o]*

*# Set the false path for the reset signal to avoid timing issues  
set\_false\_path -from [get\_ports rst\_n]*

*# Set the wire load model (assuming a medium wire load model)  
set\_wire\_load\_model -name "medium"*

<https://github.com/baruaeee/DD_Lab_exercise/blob/master/SIPO/Synthesis/IHP_Open-PDK/SDC/SIPO.sdc>

Liberty Datei kan in “/eda/cadence/pdks/sky130/sky130\_scl\_9T\_0\_0\_5/lib/” gefunden.

vorausgesetzt, die SDC, Liberty und HDL datai sind verfügber, eine TCL Datai kan geschriben und das GENUS consol kann aufrufen. Die TCL Datai ist wie folgende:

*# Loads the standard cell library for synthesis (Skywater 130nm PDK, typical corner, 1.8V, 25°C)  
read\_libs /eda/cadence/pdks/sky130/sky130\_scl\_9T\_0\_0\_5/lib/sky130\_tt\_1.8\_25\_nldm.lib*

*# Reads the Verilog HDL file for the SIPO (Serial-In-Parallel-Out) design  
read\_hdl ../../HDL/SIPO.v*

*# Elaborates the top-level module "serial\_to\_parallel" from the HDL  
elaborate serial\_to\_parallel*

*# Reads the Synopsys Design Constraints (SDC) file for timing and other constraints  
read\_sdc SDC/SIPO.sdc*

*# Configures medium effort for three synthesis stages: Generic synthesis (technology-independent), Technology mapping, Optimization  
set\_db syn\_generic\_effort medium  
set\_db syn\_map\_effort medium  
set\_db syn\_opt\_effort medium*

*# Executes the three synthesis stages in sequence  
syn\_generic  
syn\_map*

*# write the hdl file after mapping  
write\_hdl -lec > outputs/lec\_src/SIPO2mapped\_lec.v*

*# LEC do file for main rtl (golden) design vs intermediate (mapped) design  
#write\_do\_lec -golden\_design [golden/file/path==>optional if the golden is the main design] -revised\_desgn [revised\_design/file/path] -logfile <log/file/path] > [do/file/path]  
write\_do\_lec -revised\_design output/lec\_src/SIPO2mapped\_lec.v -logfile ../../Verification/Post-Synthesis/LEC/lec\_src/SIPO2mapped\_lec.log > ../../Verification/Post-Synthesis/LEC/lec\_src/SIPO2mapped\_lec.do*

*syn\_opt*

*#reports  
report\_timing > reports/report\_timing.rpt  
report\_power > reports/report\_power.rpt  
report\_area > reports/report\_area.rpt  
report\_qor > reports/report\_qor.rpt*

*#Outputs  
write\_hdl > outputs/SIPO\_netlist.v  
write\_sdc > outputs/post\_synth\_SIPO.sdc  
write\_sdf -timescale ns -nonegchecks -recrem split -edges check\_edge -setuphold split > outputs/delays.sdf  
write\_design -base\_name innovus\_src/SIPO*

*# LEC do file for intermediate (mapped) design vs Final design  
write\_do\_lec -golden\_design output/lec\_src/SIPO2mapped\_lec.v -revised\_design outputs/SIPO\_netlist.v -logfile ../../Verification/Post-Synthesis/LEC/lec\_src/mapped2final\_lec.log > ../../Verification/Post-Synthesis/LEC/lec\_src/mapped2final\_lec.do*

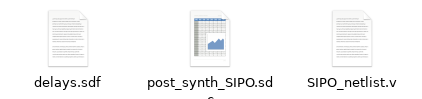
*# LEC do file for main rtl (golden) design design vs Final design  
write\_do\_lec -revised\_design outputs/SIPO\_netlist.v -logfile ../../Verification/Post-Synthesis/LEC/lec\_src/SIPO2final\_lec.log > ../../Verification/Post-Synthesis/LEC/lec\_src/SIPO2final\_lec.do*

*#exit*

Das GENUS Consol kann aufrufen mit dem folgenden (und synth.tcl Datai) Befehl aufgerufen werden:

*genus -f synth.tcl*

Es werden alle erforderlichen Daten für den nächsten Schritt erzeugt.



LEC (Logic Eqivalent Check):