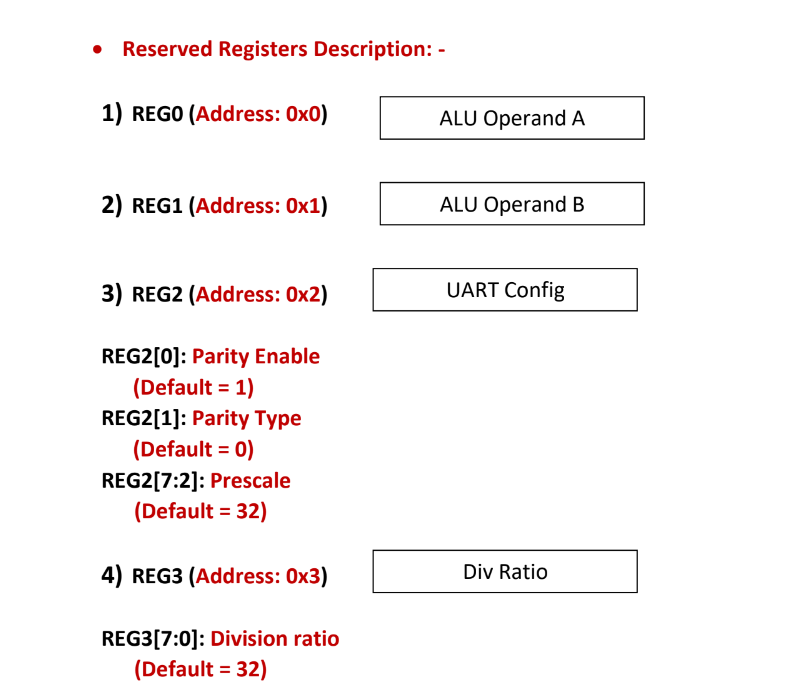
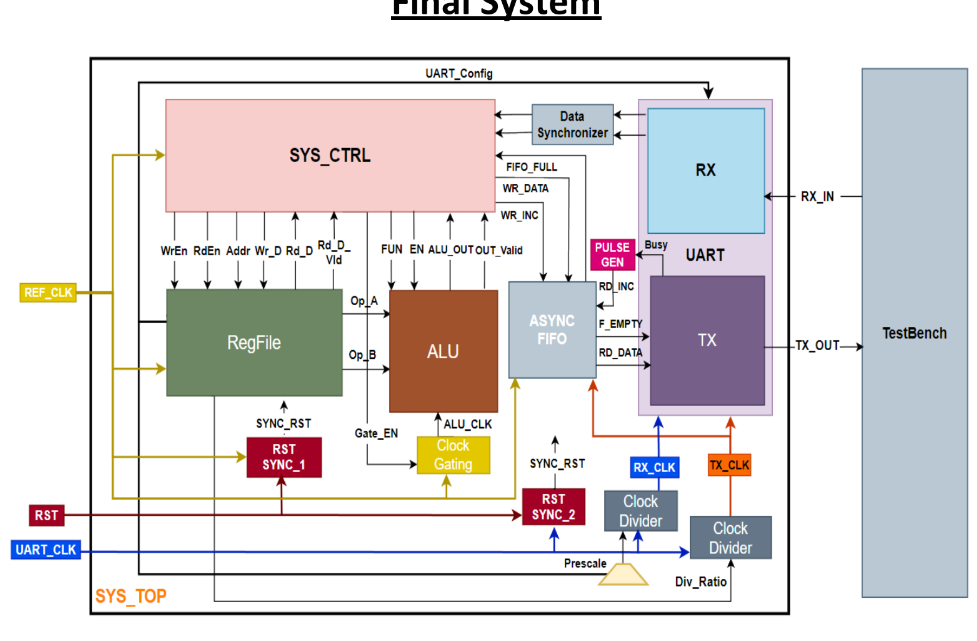
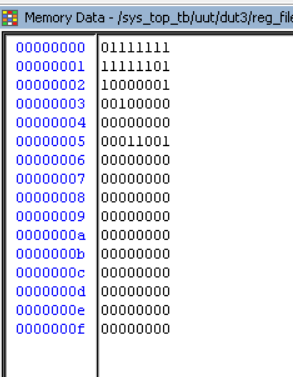


multiple clock low power system RTL to GDS

System specs

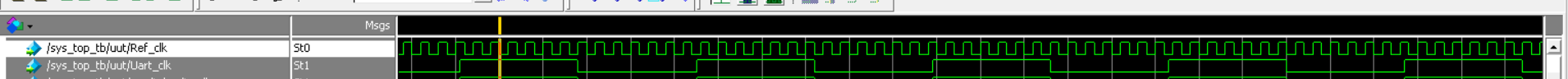
 

Test bench snippets for some of the test cases



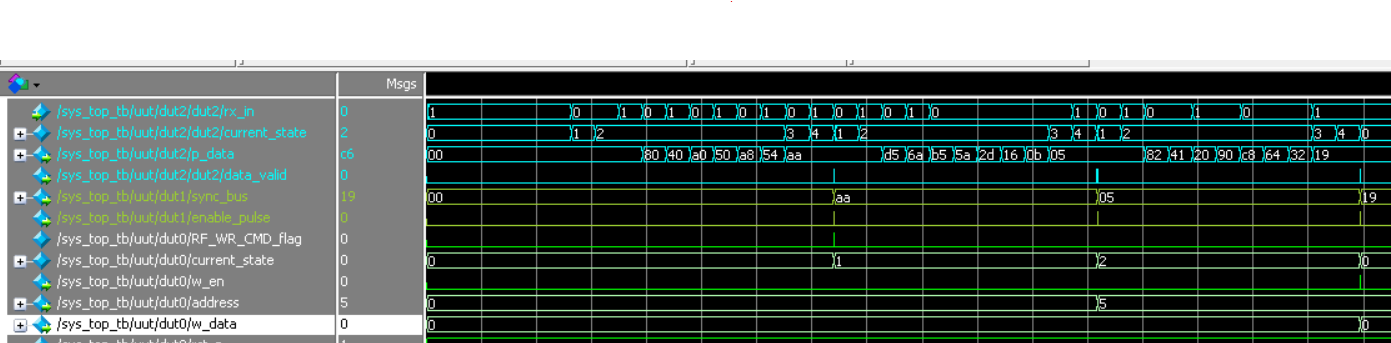
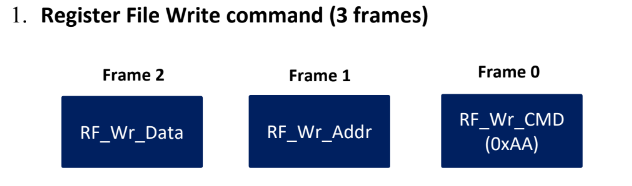
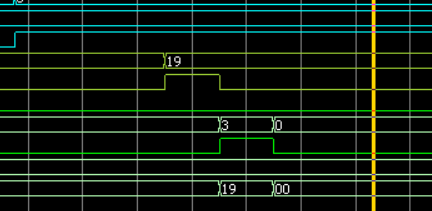
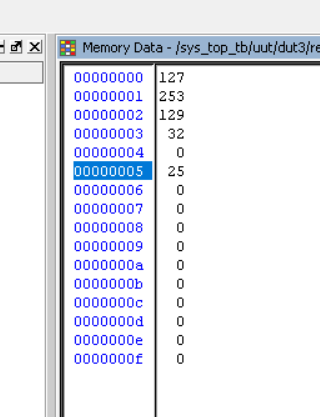
Reserved registers data is saved correctly as mentioned in the specs

Clock relations





Gated clk is on when the alu is doing its operation but it is turned on before the alu calulates the result to be stable and ready



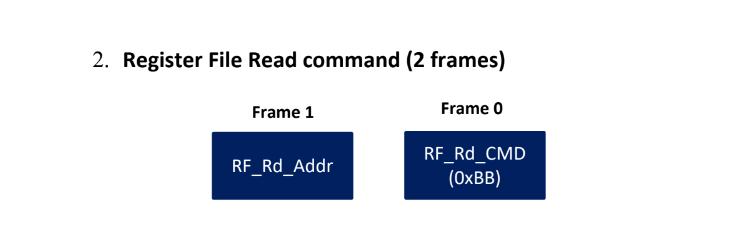
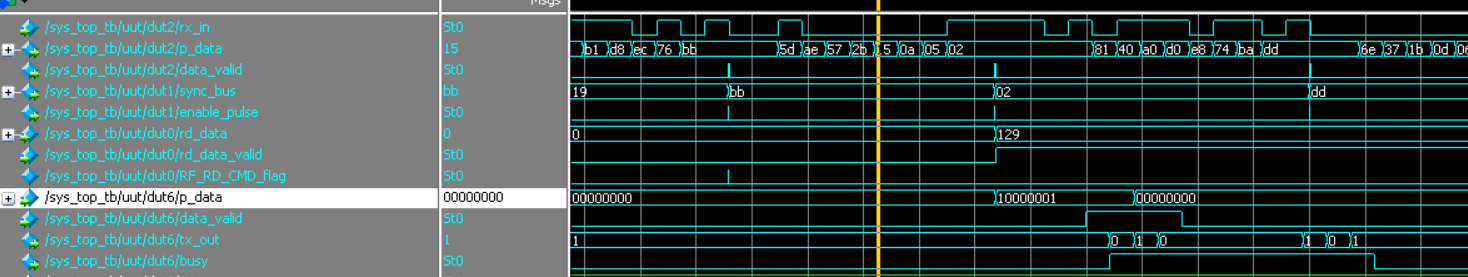
Data is stored in address 5 in reg file as expected

Output of data synchronizer between the uart rx and system control :

Command code = 0xaa in this state the RF\_WR\_CMD\_FLAG is set to high

write address = 0x05 in this state the address is registered until the write data arrives in the next data

write data = 0x19 (8’d25) in this state the sys\_ctrl set the w\_en of the reg file to high then stores the data in address 5 in the reg\_file

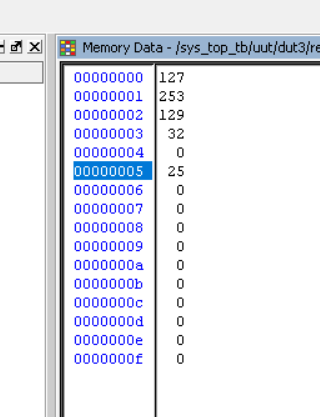


Readed data from the register file is as expected (8’d129)

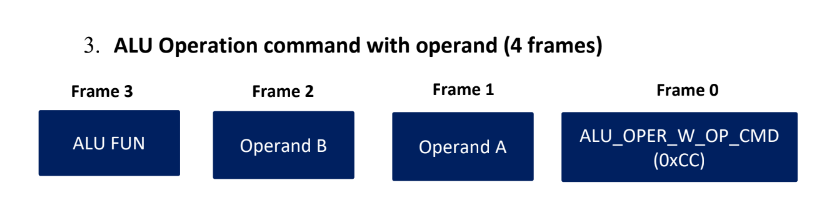
Output of data synchronizer between the uart rx and system control :

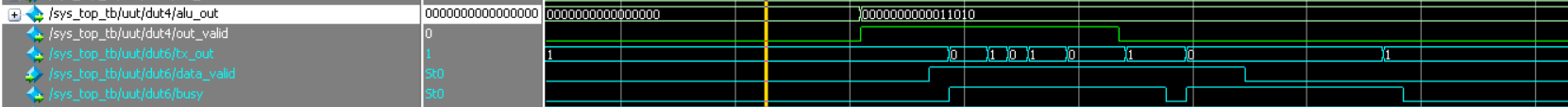
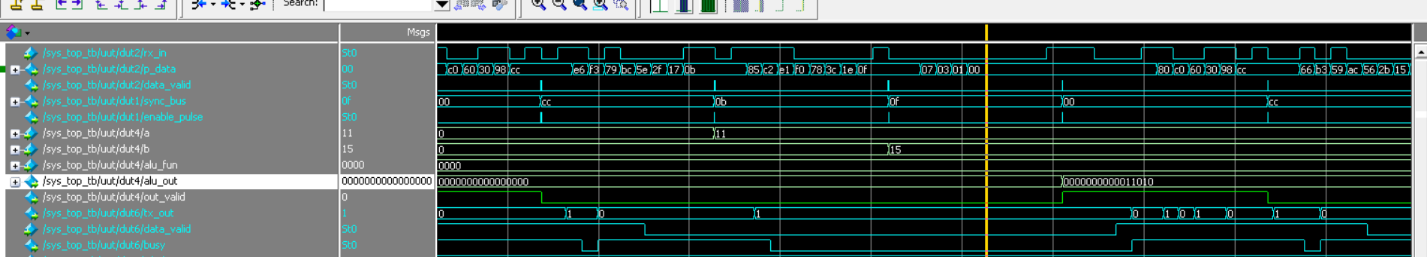
Command code = 0xbb in this state the RF\_RD\_CMD\_FLAG is set to high

read address = 0x02



* The readed data is then stored in the fifo until the uart tx read it and transmit it as follows :start bit , data bits , even parity bit , stop bit and the busy signal is high as long as the frame is being sent
* the data valid signal of the tx is set to high when there is any data stored in the fifo





* The output of the alu is 16 bits and the fifo data width is 8 bits so the fifo stores any data comes from the alu in two 8-bits locations then the the uart tx transmits the 8-LSB first in a single frame then transmits the 8-MSB in another single frame (each frame of them has start , data , parity and stop bits)
* Busy signal is high from the start state to the stop state and then goes low in idle state

another single frame (each frame of them has start , data , parity and stop bits)

Output of alu is as expected

(11 +15 = 25)

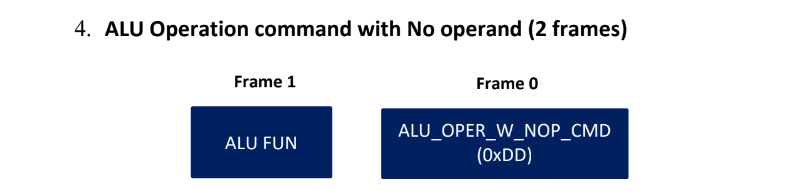
Output of data synchronizer between the uart rx and system control :

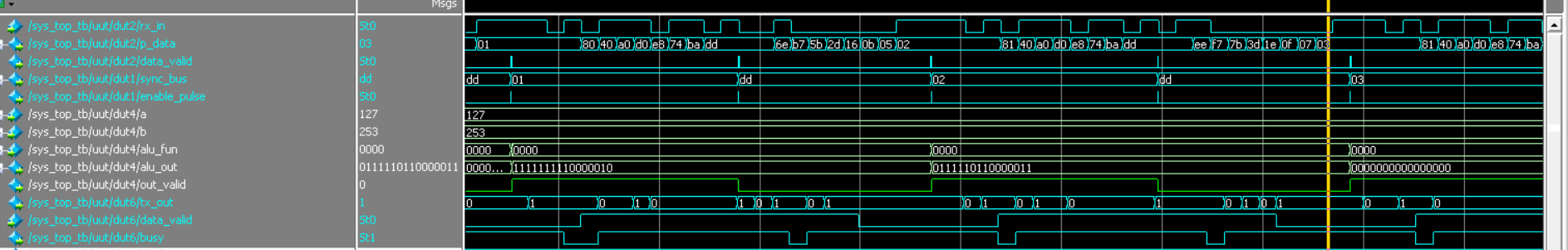
Command code = 0xcc

Operand a value = 0x0b (stored in address 0 in reg file by default)

Operand b value = 0x0f (stored in address 1 in reg file by default)

Alu\_fun value = 0x00 (do addition)





* The output of the alu is 16 bits and the fifo data width is 8 bits so the fifo stores any data comes from the alu in two 8-bits locations then the the uart tx transmits the 8-LSB first in a single frame then transmits the 8-MSB in another single frame (each frame of them has start , data , parity and stop bits)
* Busy signal is high from the start state to the stop state and then goes low in idle state

Output of alu is as expected

(127\*253 = 32131)

Output of data synchronizer between the uart rx and system control :

Command code = 0xdd

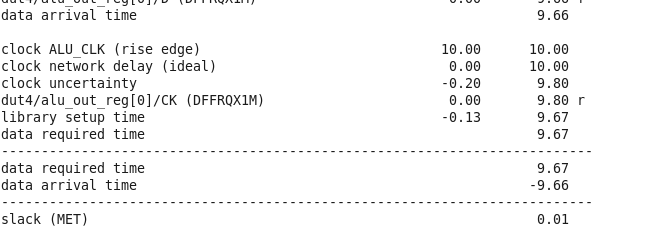
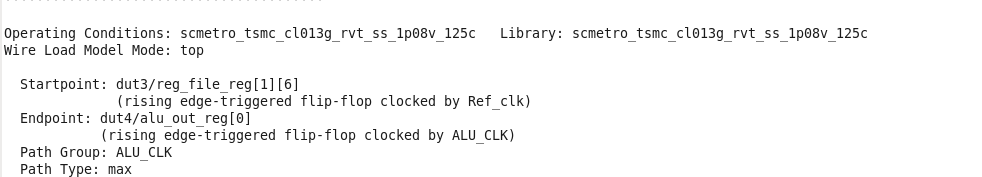
Operand a value = this command reads the value stored in address 0 in the reg file which is 8’d127

Operand b value = this command reads the value stored in address 1 in the reg file which is 8’d253

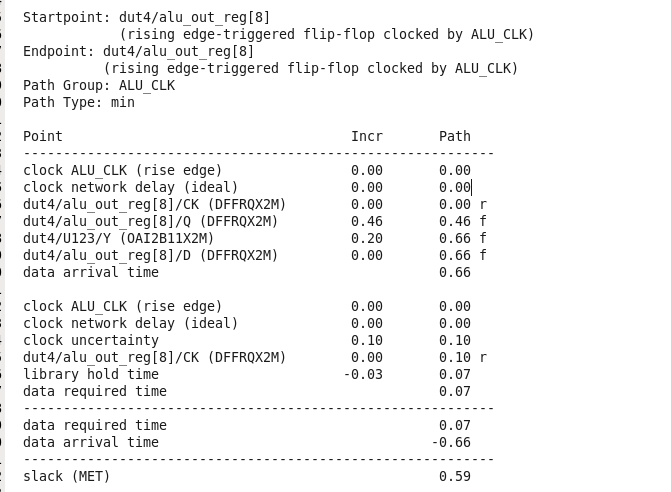
Alu\_fun value = 0x02 (do multiply)

synthesis

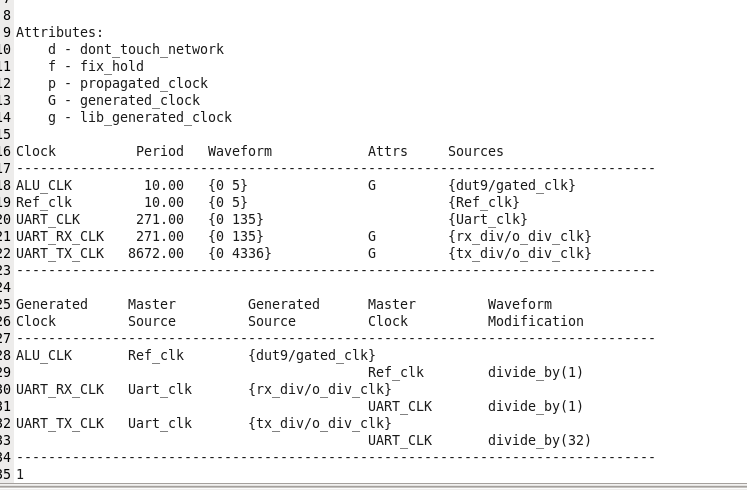
Worst setup path



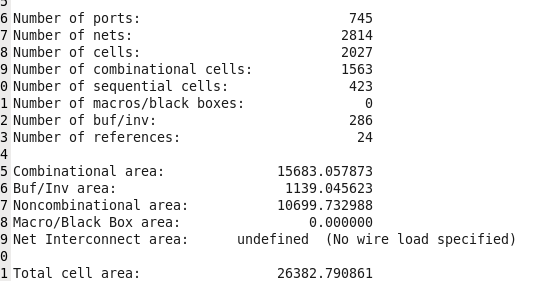
Worst hold path



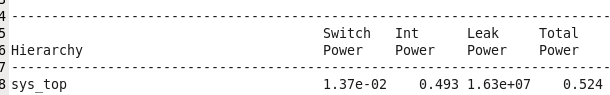
Master and generated clocks



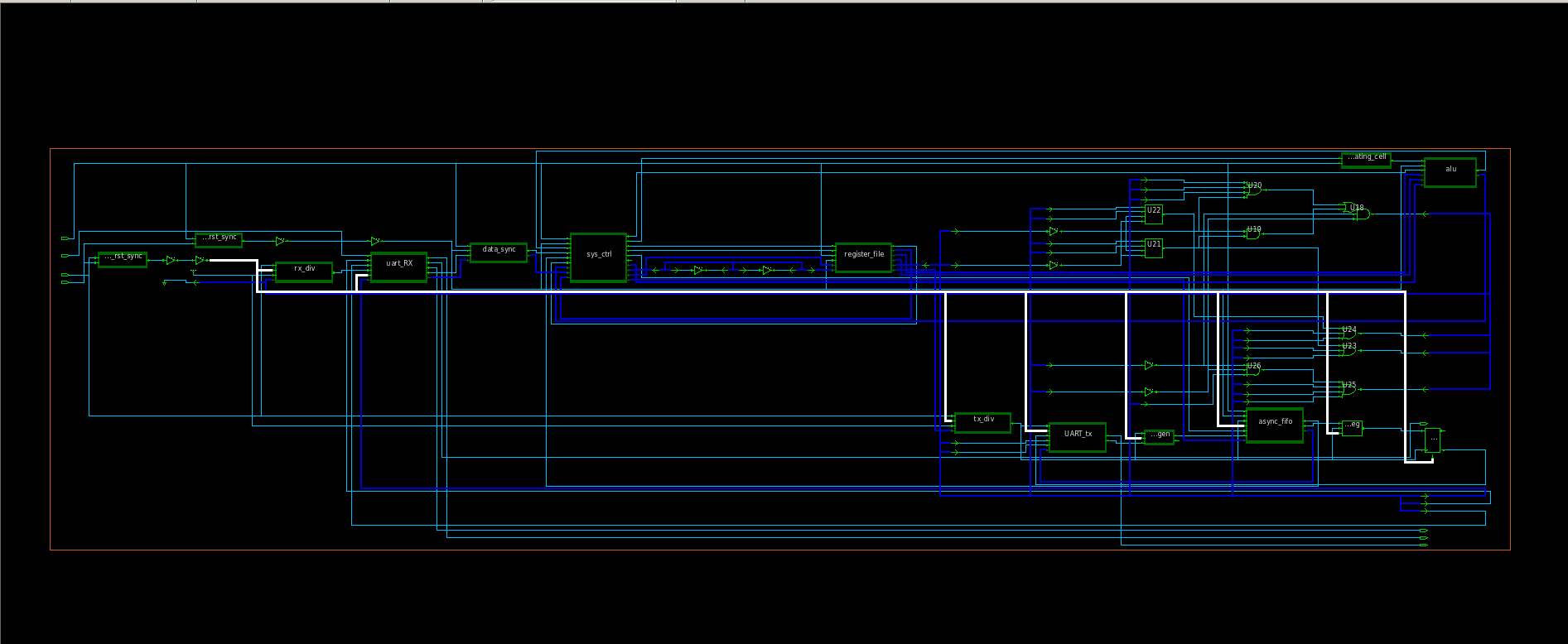
Total area



Total power

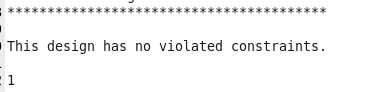
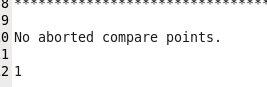


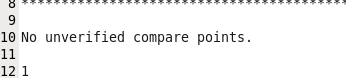
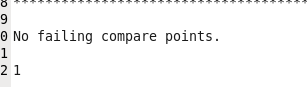
Synthesis gate level netlist snippets



Post synthesis formality



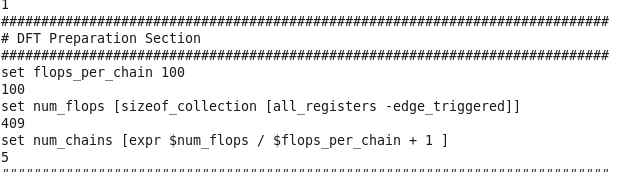
 



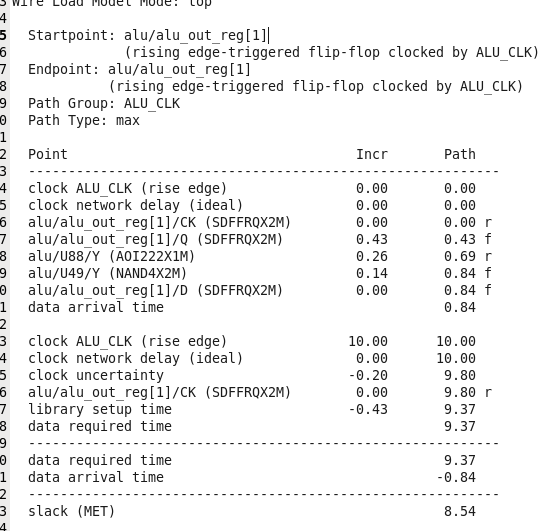


DFT

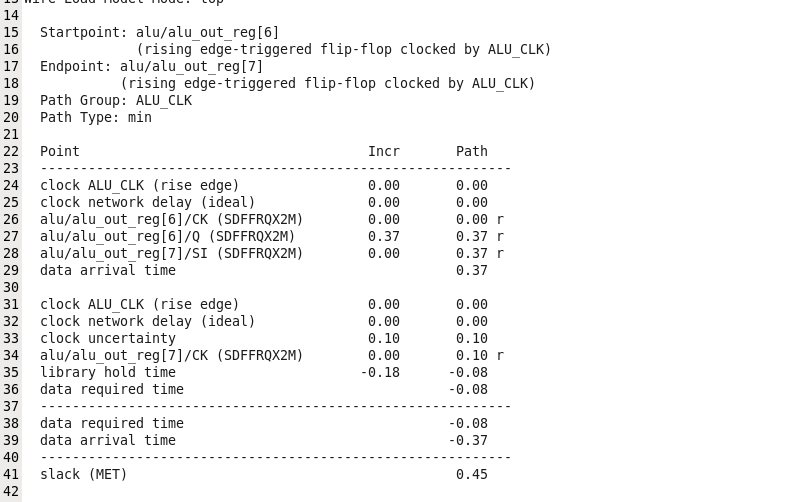
Calculate number of chains



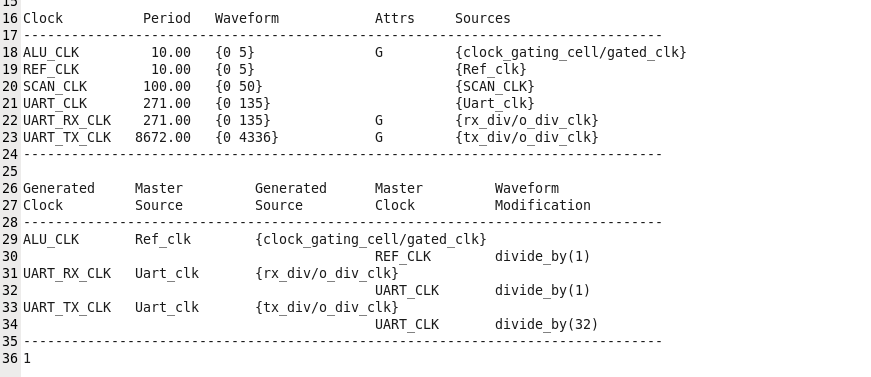
Worst setup path



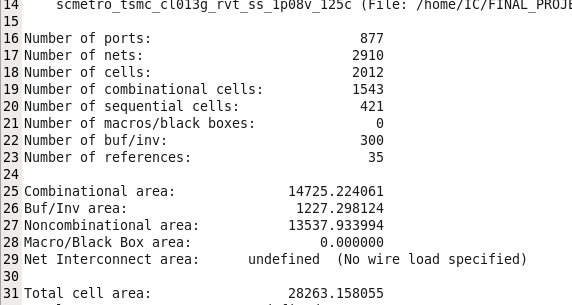
Worst hold path



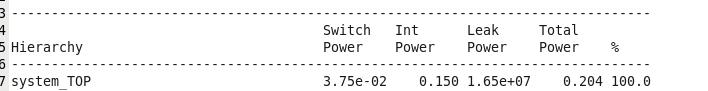
Master and generated clocks



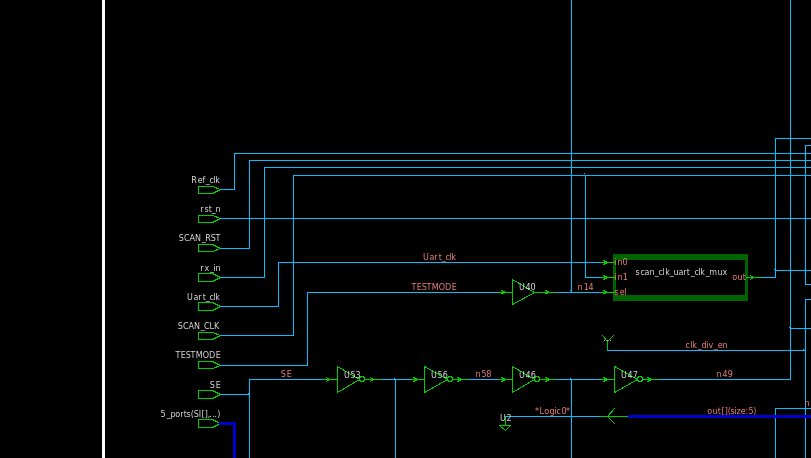
Total area

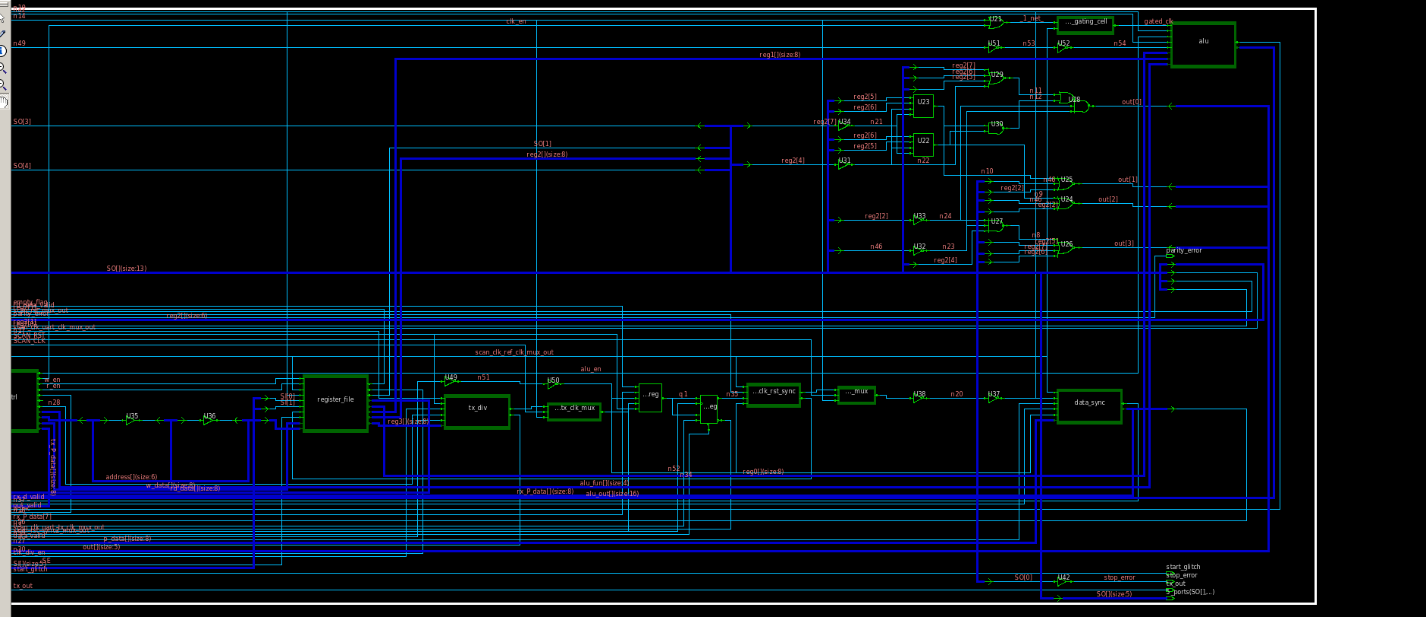
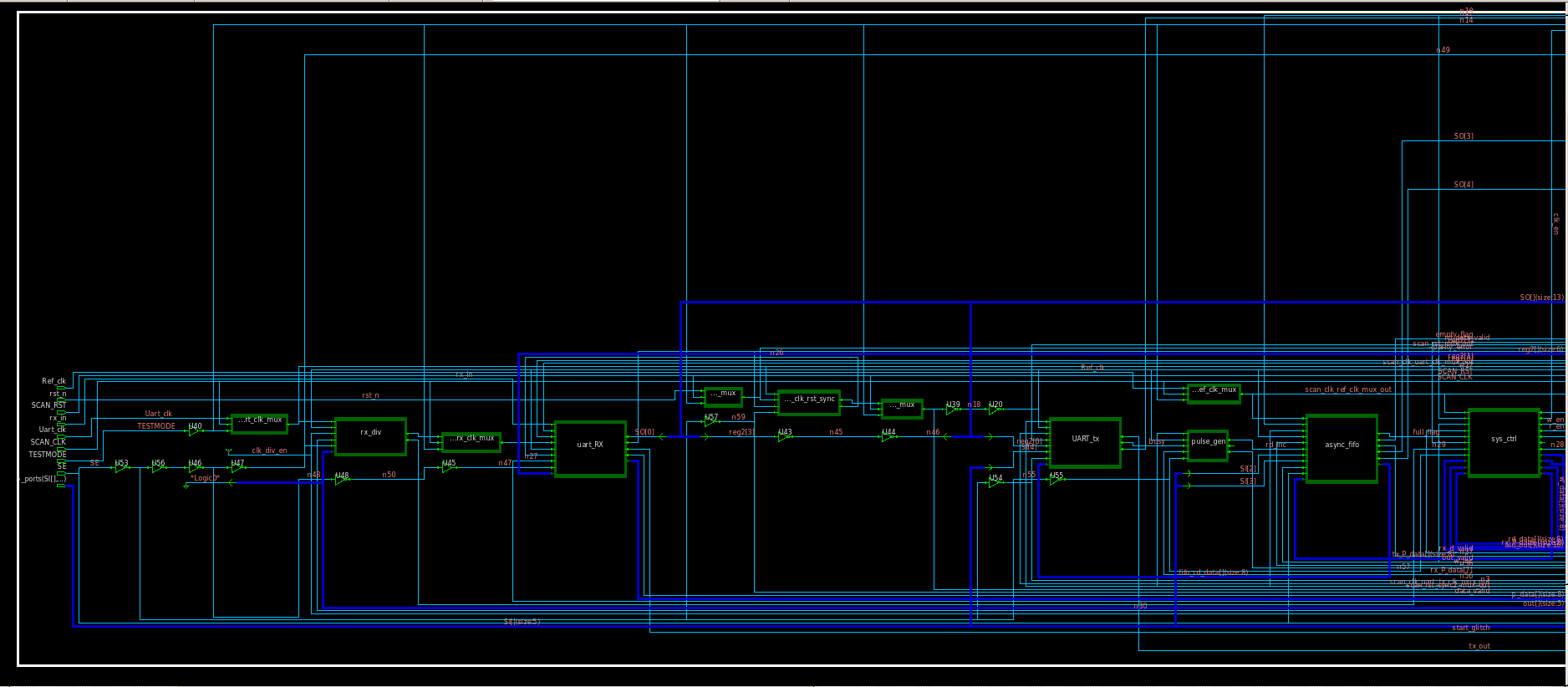
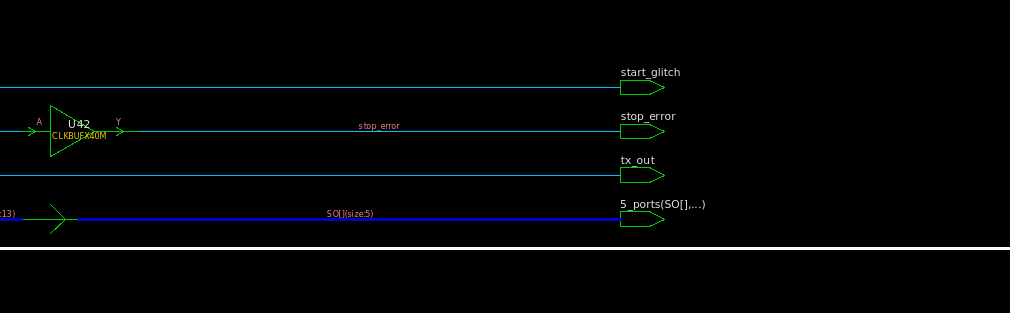


Total power

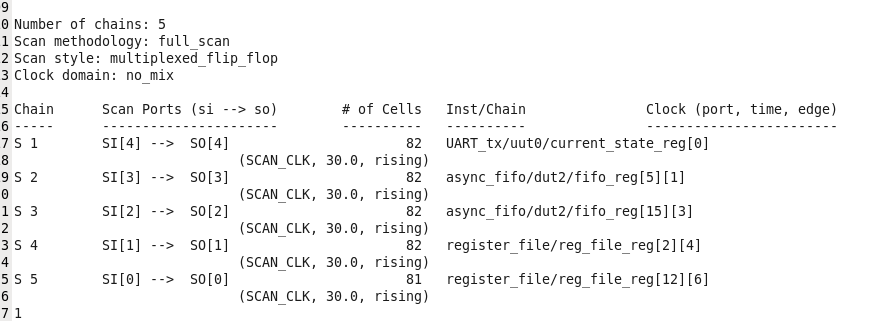


DFT gate level netlist snippets

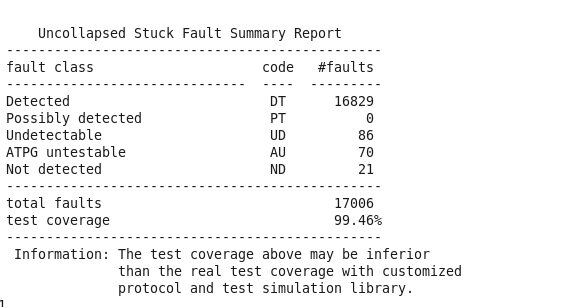




Scan chain distribution

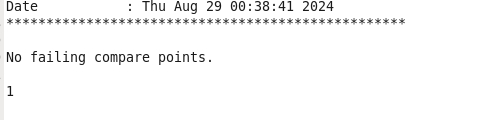
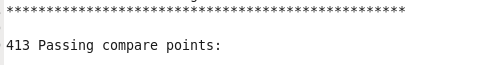
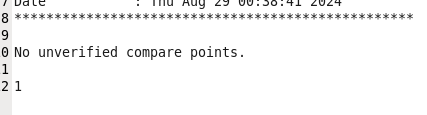
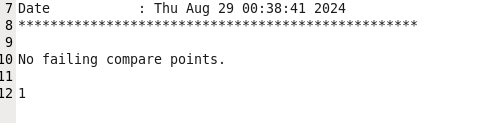
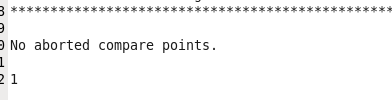
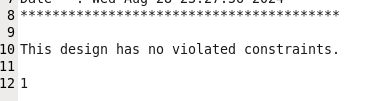


Coverage percentage



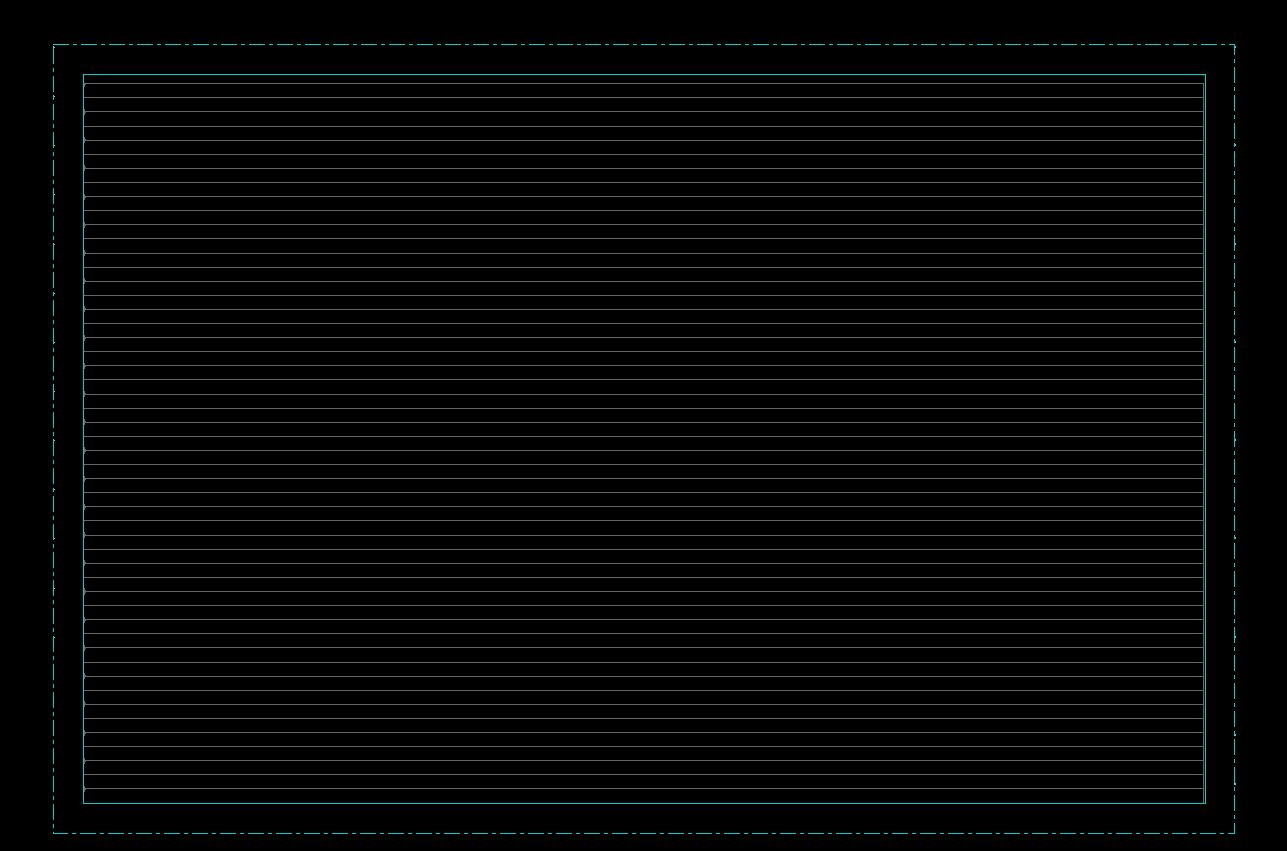
Post dft formality

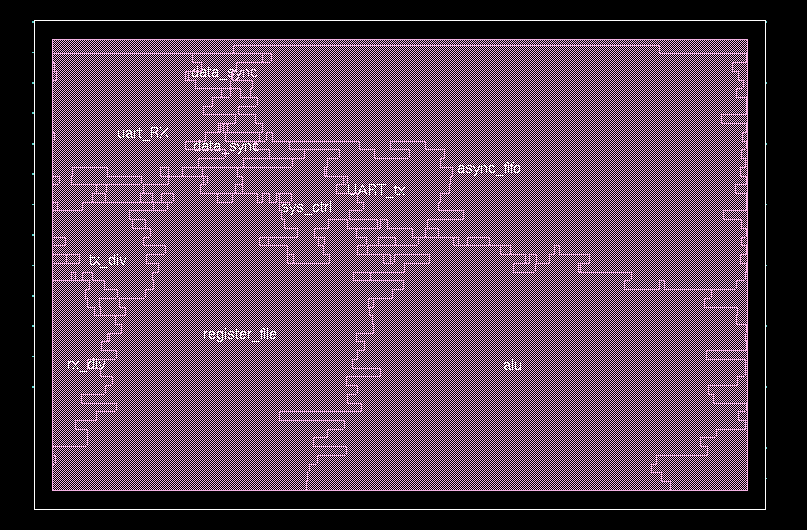




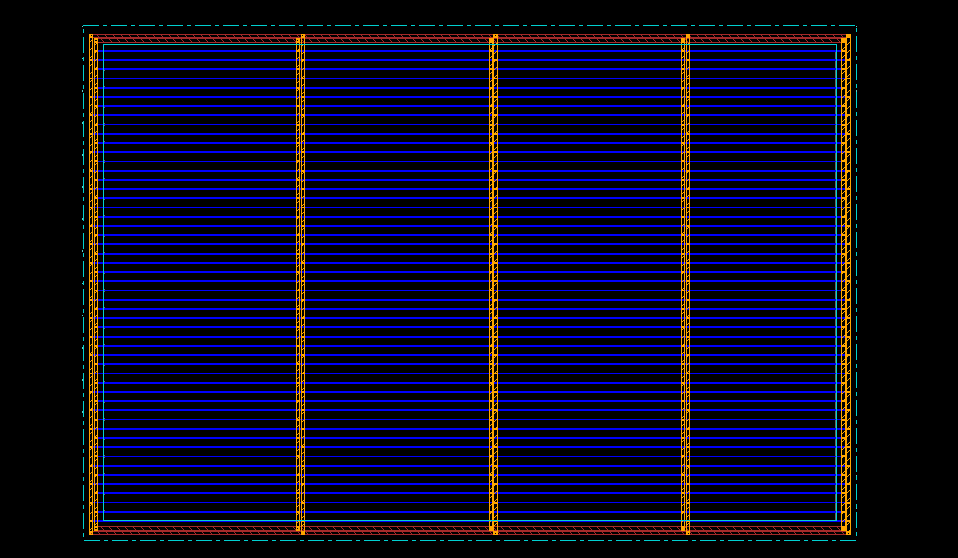
PNR

Floor planning

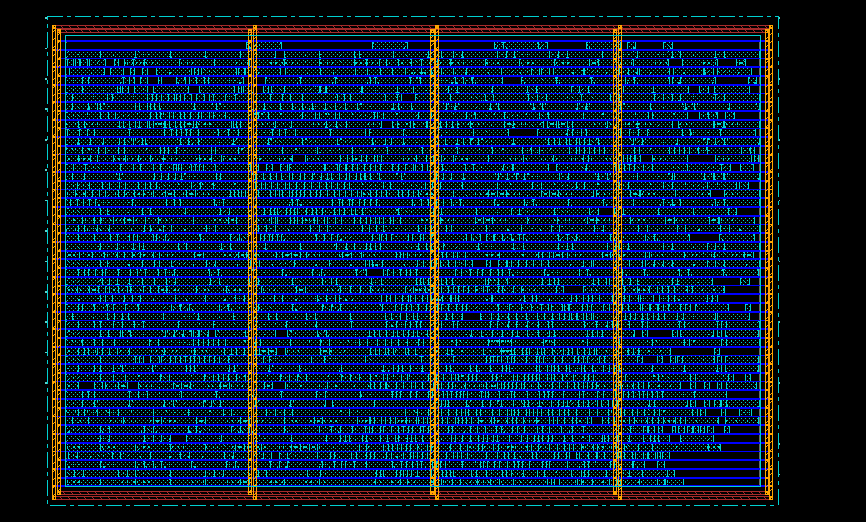




Power planning



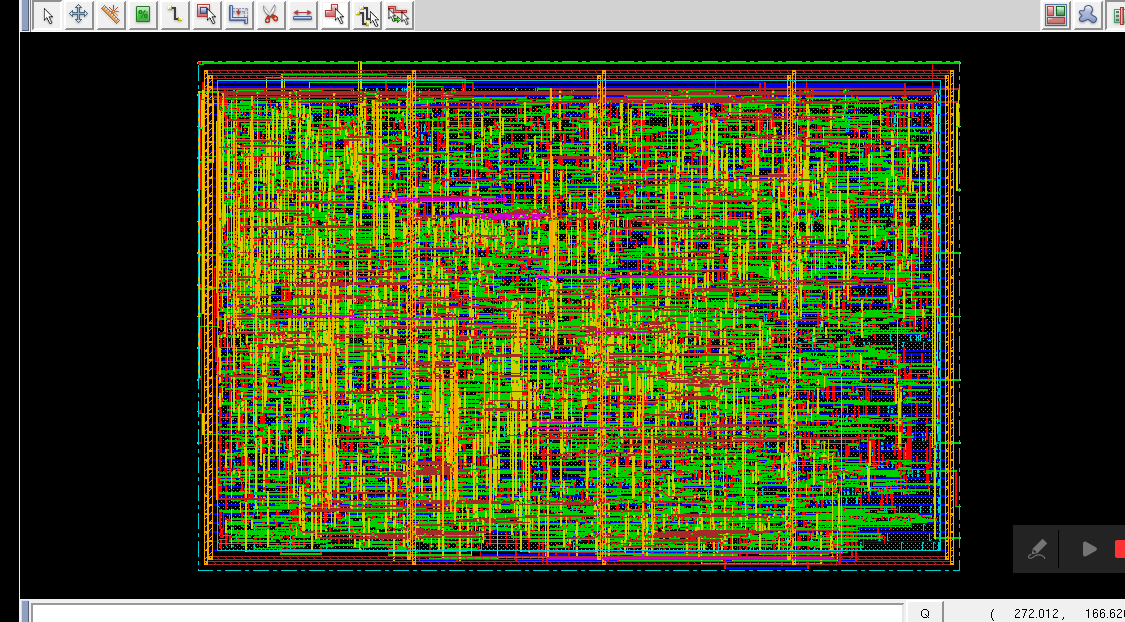
Placement



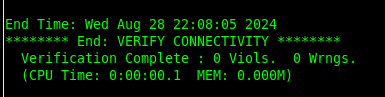
Add filler cells

# 

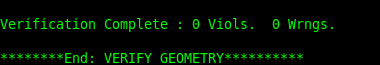
Routing (post route)



Check connectivity



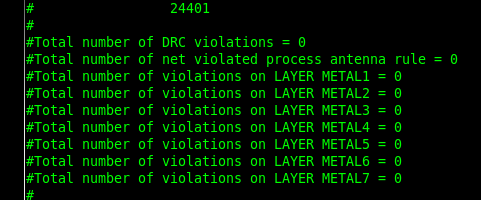
Check geometry



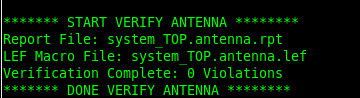
Check if there are any gaps after adding filler cells



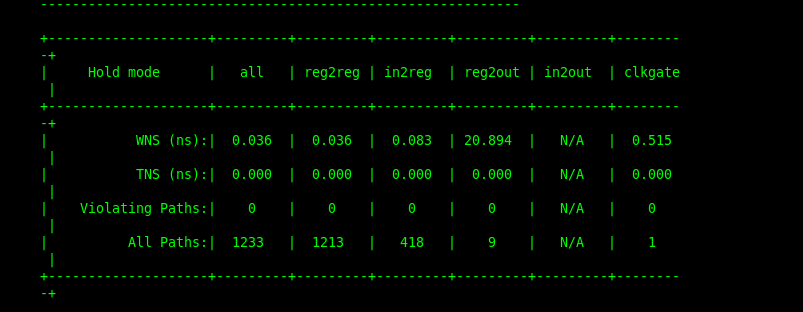
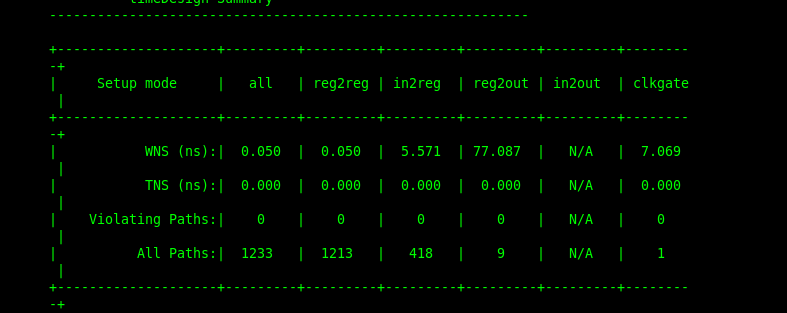
Check drc violations



Check process antenna



Setup and hold timing analysis (post route)



* **Link for the rtl code of the system and all backend scripts and files :** [**github link**](https://github.com/Omarafifi1/multiple-clock-low-power-system-RTL-to-GDS-)
* **Link for the rtl code and functional verification of each block :[github link](https://github.com/Omarafifi1/ASIC-IC-DESIGN-COURSE-PROJECTS/tree/main/VERILOG/signed_16_bit_alu_with_new_added_features)**