Four Port Switch Project

- Leen Zoabi 212930366
- Yazan Atamna 325072551

Index:

Stage A

1. Introduction:

The main goal of our project is to develop a complete Register Transfer Level (RTL) design of for a simplified four-port packet switch. This design must correctly handle packet reception, arbitration and forwarding to one or more destination ports based on the packet's embedded fields.

The key objectives for this stage were:

- Implementing a robust and modular RTL design in Verilog/SystemVerilog .
- Ensuring proper clock and reset handling , adhering to synchronous design principles.
- Implementing essential flow control logic, including a finite state machine (FSM) and arbitration to handle concurrent packet arrivals.
- Validating basic functionality using initial Quality Assurance (QA) test cases.

2. Logical Design:

2.1:

1) **switch_port Module (per-port Logic**) : Each physical port from port0 to port3 is connected to its own instance of switch_port. This module contains the local FSM to manage packet ingestion, latency (using FIFO registers) and preparation of the packet for routing across the switch fabric.

2) **switch_4port Module (Central Crossbar/Arbitration):** this top-level module is responsible for arbitration and broadcast replication.

- It collects the routed packet data from all four instances
- An arbitration logic block (using an always_comb) selects one packet from all valid inputs using Round Robin scheme.
- A crossbar logic block (implemented using an always_ff block) is triggered by the arbitration winner, it checks the winning's packet's target mask and replicates the packet data to all specified destination output ports, the outputs are registered for stability.

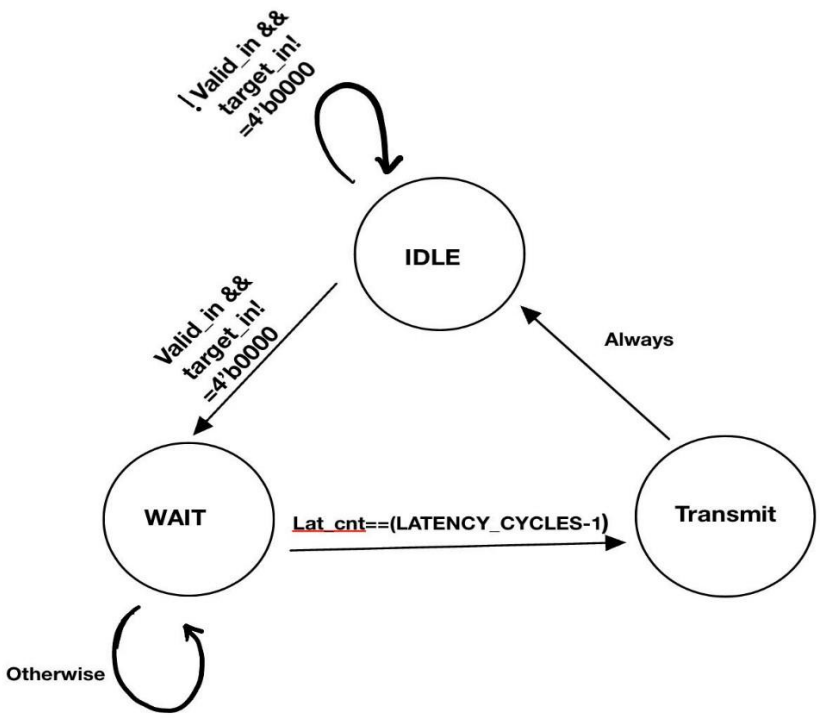
2.2 Per-Port State Machine (FSM Diagram)

The core control logic for packet ingestion and holding within each switch_port instance is managed by a synchronous Finite State Machine (FSM). The FSM uses the state t enumeration, consisting of three states:

1) IDLE: the default state , waiting for a valid incoming packet.

2) WAIT_LAT: a latency state introduced after packet acceptance , delaying the packet's availability for arbitration. The duration is fixed by LATENCY_CYCLES (set to 5).

3) TRANSMIT: the state where the packet is marked as valid for selection by the central arbitration unit.



graph 1 - state diagram

State diagram explanation:

Current State	Transition Condition	Next State	Action/Output
IDLE	valid_in && (target_in != 4'b0000)	WAIT_LAT	Packet is captured (accepted).
IDLE	!(valid_in && (target_in != 4'b0000))	IDLE	The state remains IDLE
WAIT_LAT	lat_cnt == (LATENCY_CYCLES-1)	TRANSMIT	Latency count expires.
WAIT_LAT	Otherwise	WAIT_LAT	Increment internal latency counter.
TRANSMIT	Always	IDLE	The port releases the packet (single-cycle transmit model). If a new packet is immediately arriving, it is accepted and overrides the consumption of the current packet.

2.3 Reset and Clockin Strategy

The switch design follows synchronous design rules and uses non blocking assignments (<=) in all sequential blocks to correctly represent hardware registers.

Clocking: the system clock(clk) is the primary timing refrence, triggering all sequential logic on the positive edge (@(posedge clk))

Reset:

- A single active-low asynchronous reset signal (rst_n) is used for initialization.
- The Verilog implementation models asynchronous reset assertion with synchronous reset de-assertion. Which ensures :

1- Fast reset assertion: immediate reset regardless of clock edge
2- Metastability avoidance: the reset signal's de-assertion is synchronized to the clock domain , which prevents the flip flop's output from entering a metastable state.

• The reset logic forces all sequential registers (state, fifo_valid, counters, output port registers in switch_4port) to an unknown cleared state (IDLE,1`b0,4`b0 and 8`h00)

2.4 Arbitration and Routing Logic

A- Arbitration (Round_Robin): to handle packet arrivals on multiple input ports at the same time, Round robin scheme was implemented in order to select one winning packet at a time.

- rr_ptr – a 2 bit register which acts like a pointer by storing the id of the port that was selected in the previous successful cycle.
- The arbitration logic iterates through all 4 ports starting from the port immediately after the rr_ptr modulo 4.
- The first port encountered with an asserted valid signal wins the arbitration and its source id is captured as the chosen.
- When a packet is selected , the rr_ptr is updated to point to the next port for the subsequent cycle , ensuring fairness.

B- Routing : the routing is implemented in the primary always_ff block of switch_4port using a registered crossbar approach.

- The chosen packet data is replicated to all output ports at the same time.
- For a packet received from chosen_src its destination is determined by target:
  1) The target is interpreted as a mask where each bit corresponds to a destination port.
  2) If a bit N is set in the target mask , the winning packet is forwarded to output port N .

3) This successfully implements single destination, multicast , and broadcast routing as required.

3. Stage A results and verification:

In order to confirm our work, we included a quality assurance QA tests , executed in the stage_a_test.sv testbench. These test validate basic functionality, ensure correct compilation and make sure our work meets design verification in order to move on to the next stages.

3.1 Testbench structure:

the QA testbench is a non object oriented verification environment using simple Verilog.

- Clock and reset: the testbench generates clock signal (clk) and drive the asynchronous reset signal (rst_n).
- Interfaces: it instantiates the DUT switch_4port connecting it to four instances of the system Verilog port-if interfaces.
- Stimulus tasks: tasks such as drive_packet simplify test case creation by abstracting the setting of input signals (source_in , target_in and data_in) for a single clock cycle injection.
- Checking tasks: the wait_for_valid tast and explicit if/e;lse checks compare the D'T's outputs in order to decide whether it's a pass or fails situation.

3.2 QA Simulation Log

The whole simulation log is attached to its own file in the submission files.

```
--- Stage A QA Testbench START ---

[TEST 1] Single Destination: P0 -> P1
[PASS] P0 -> P1 single-destination OK

[TEST 2] Multicast: P0 -> P1 & P2
[PASS] Multicast routing OK

[TEST 3] Broadcast: P0 -> All
[PASS] Broadcast routing OK

[TEST 4] Single Destination: P2 -> P3
[PASS] Non-P0 source routing OK

[TEST 5] Reset Behavior
[PASS] Outputs low during reset
[PASS] Reset clears outputs

--- Stage A QA Testbench FINISHED ---
```

4. Conclusion :

Stage A is complete. The RTL design divided into the top module switch_4port and four switch_port instances, meets all foundational requirements for modular design, synchronous clock/reset handling and basic multi destination routing.

The QA tests confirm the essential functionality of the switch core for single destination, multicast and broadcast. Which gives s a stable foundation for the next stage.

Stage B

1- Introduction:

Stage B implements a modular, object-oriented verification environment for the Four-Port Switch. This environment utilizes Constrained-Random Stimulus Generation and a layered architecture (Sequencer, Driver, Monitor, Agent) to validate the RTL design developed in Stage A.

2- Verification Architecture:

Each port is managed by a dedicated Packet Verification Component which is Packet VC

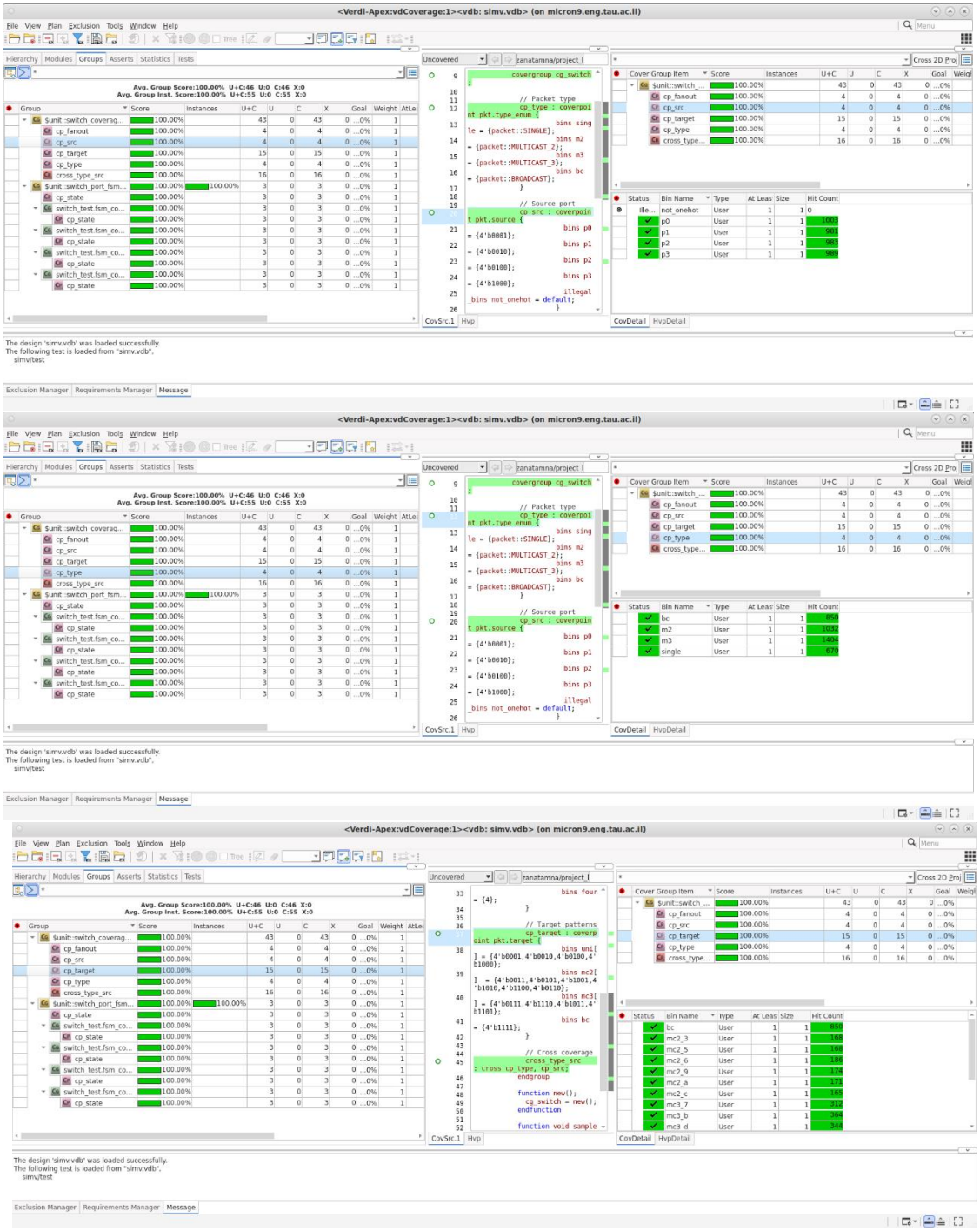
- Component base class: all verification components inherit from the component_base providing a uniform naming convention and path management via the pathname () method.
- Packet model: the packet class contains rand fields for data and target. It includes a post_randomize() function to determine the type_enum and static packet_count to trac object creation.
- Sequencer: generates traffic patterns by randomizing packet types based on a distribution and applying specific constraints to the target field.
- Driver: interfaces with port_if via a virtual interface. The driver uses a static semaphore (drive_sem) to manage serialized access and prevent signal contention during packet injection
- Monitor: passively observes the output side of the port_if. It uses the collect_packet task to reconstruct packet objects from the physical signals and sends them to the checker via a mailbox.
- Agent and Packet vc : the agent encapsulates the sequencer, driver and monitor while the packet vc provides top level configuration for each port, assigning the virtual interfaces and port index.

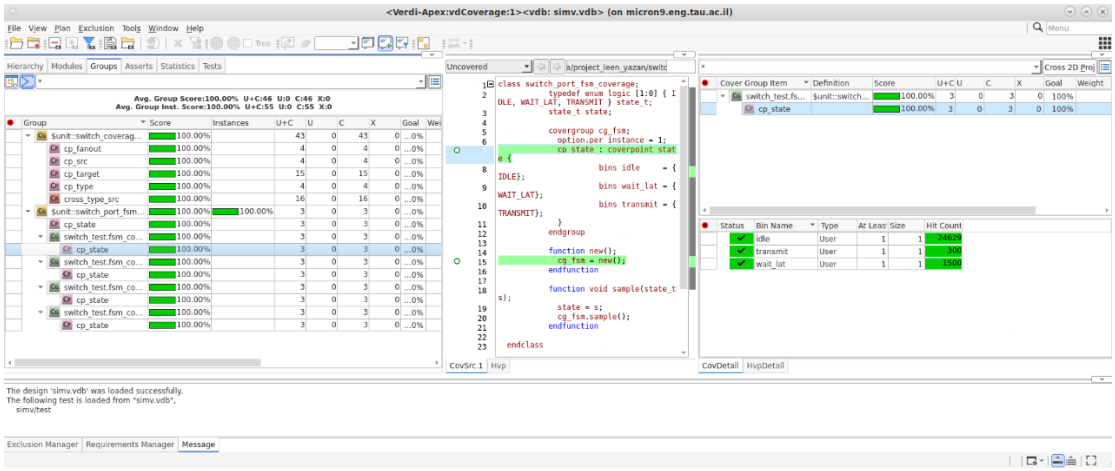
3- Checker implementation: the checker class is crucial for this stage and it is the heart of the validation.

- Maintains four expected “mailboxes” from the driver and four actual ones from the monitor
- When a packet is driven, the checker clones it and puts it in a exp_list which is a queue for every destination bit set in the target mask.
- Validation: the check_port continuously compares monitored packets against the expected queue. and it makes sure data remains uncorrupted and that packets arrive at the correct ports.
- When the simulation is complete report_and_asserts calculates pass/fail and ensures no packets are missing (stuck in the switch)

4- **Coverage**: in order to validate that the verification plan is fulfilled , we implement functional coverage in switch_test

- Tracks type_enum , source port and target fanout, FSM states of the switch_port.
- Tracks combinations of source ports and packet types to ensure the switch was tested under all routing scenarios.
  Coverage results:





5- Simulation Results:

The test confirmed that the round robin/ fifo logic handled simultaneous arrivals

The results:

```
[CHK] PASS dst3: Src=0100 Tgt=1111 Data=c3 Time=262885
[vc0.agent.seq] Generated: Src=0001 Tgt=1110 Data=54 Type=MULTICAST_3
[vc0.agent.drv] DRIVE: Src=0001 Tgt=1110 Data=54 Type=MULTICAST_3 Time=263015
[vc1.agent.mon] MON:   Src=0001 Tgt=1110 Data=54 Type=MULTICAST_3
[vc2.agent.mon] MON:   Src=0001 Tgt=1110 Data=54 Type=MULTICAST_3
[vc3.agent.mon] MON:   Src=0001 Tgt=1110 Data=54 Type=MULTICAST_3
[CHK] PASS dst1: Src=0001 Tgt=1110 Data=54 Time=263105
[CHK] PASS dst2: Src=0001 Tgt=1110 Data=54 Time=263105
[CHK] PASS dst3: Src=0001 Tgt=1110 Data=54 Time=263105
[vc1.agent.seq] Generated: Src=0010 Tgt=0001 Data=b2 Type=SINGLE
[vc1.agent.drv] DRIVE: Src=0010 Tgt=0001 Data=b2 Type=SINGLE Time=263235
[vc0.agent.mon] MON:   Src=0010 Tgt=0001 Data=b2 Type=SINGLE
[CHK] PASS dst0: Src=0010 Tgt=0001 Data=b2 Time=263325
[vc3.agent.seq] Generated: Src=1000 Tgt=1100 Data=f8 Type=MULTICAST_2
[vc3.agent.drv] DRIVE: Src=1000 Tgt=1100 Data=f8 Type=MULTICAST_2 Time=263455
[vc2.agent.mon] MON:   Src=1000 Tgt=1100 Data=f8 Type=MULTICAST_2
[vc3.agent.mon] MON:   Src=1000 Tgt=1100 Data=f8 Type=MULTICAST_2
[CHK] PASS dst2: Src=1000 Tgt=1100 Data=f8 Time=263545
[CHK] PASS dst3: Src=1000 Tgt=1100 Data=f8 Time=263545
[vc2.agent.seq] Generated: Src=0100 Tgt=1100 Data=8b Type=MULTICAST_2
[vc2.agent.drv] DRIVE: Src=0100 Tgt=1100 Data=8b Type=MULTICAST_2 Time=263675
[vc2.agent.mon] MON:   Src=0100 Tgt=1100 Data=8b Type=MULTICAST_2
[vc3.agent.mon] MON:   Src=0100 Tgt=1100 Data=8b Type=MULTICAST_2
[CHK] PASS dst2: Src=0100 Tgt=1100 Data=8b Time=263765
[CHK] PASS dst3: Src=0100 Tgt=1100 Data=8b Time=263765
[vc0.agent.seq] Generated: Src=0001 Tgt=1110 Data=4b Type=MULTICAST_3
[vc0.agent.drv] DRIVE: Src=0001 Tgt=1110 Data=4b Type=MULTICAST_3 Time=263895
[vc1.agent.mon] MON:   Src=0001 Tgt=1110 Data=4b Type=MULTICAST_3
[vc2.agent.mon] MON:   Src=0001 Tgt=1110 Data=4b Type=MULTICAST_3
[vc3.agent.mon] MON:   Src=0001 Tgt=1110 Data=4b Type=MULTICAST_3
[CHK] PASS dst1: Src=0001 Tgt=1110 Data=4b Time=263985
[CHK] PASS dst2: Src=0001 Tgt=1110 Data=4b Time=263985
[CHK] PASS dst3: Src=0001 Tgt=1110 Data=4b Time=263985

==== CHECKER SUMMARY ====
EXPECTED_TOTAL=2756  RECEIVED=2756  PASS=2756  FAIL=0  MISSING=0  EXP_MB_PENDING=0
### SWITCH_TEST FINISHED at time 264315 ###
$finish called from file "switch_test.sv", line 113.
$finish at simulation time              264315
           V C S   S i m u l a t i o n   R e p o r t
Time: 264315
CPU Time:      0.590 seconds;       Data structure size:   0.1Mb
Wed Jan  7 09:49:34 2026
#qrsh -V -cwd -b y -q sim_normal ./simv +ntb_random_seed_automatic 2>&1 | tee log
```

Stage C

1- **Introduction:** The transition from Stage B to Stage C shows the shift from functional verification to hardware realization. While the previous stage ensured the RTL logic was functionally correct through testing, Stage C focuses on logic synthesis which converts the SystemVerilog code into a gate level netlist which is optimized for physical implementation. The main goal is to transform the model of the four-port switch into a structural representation that meets specific hardware constraints:

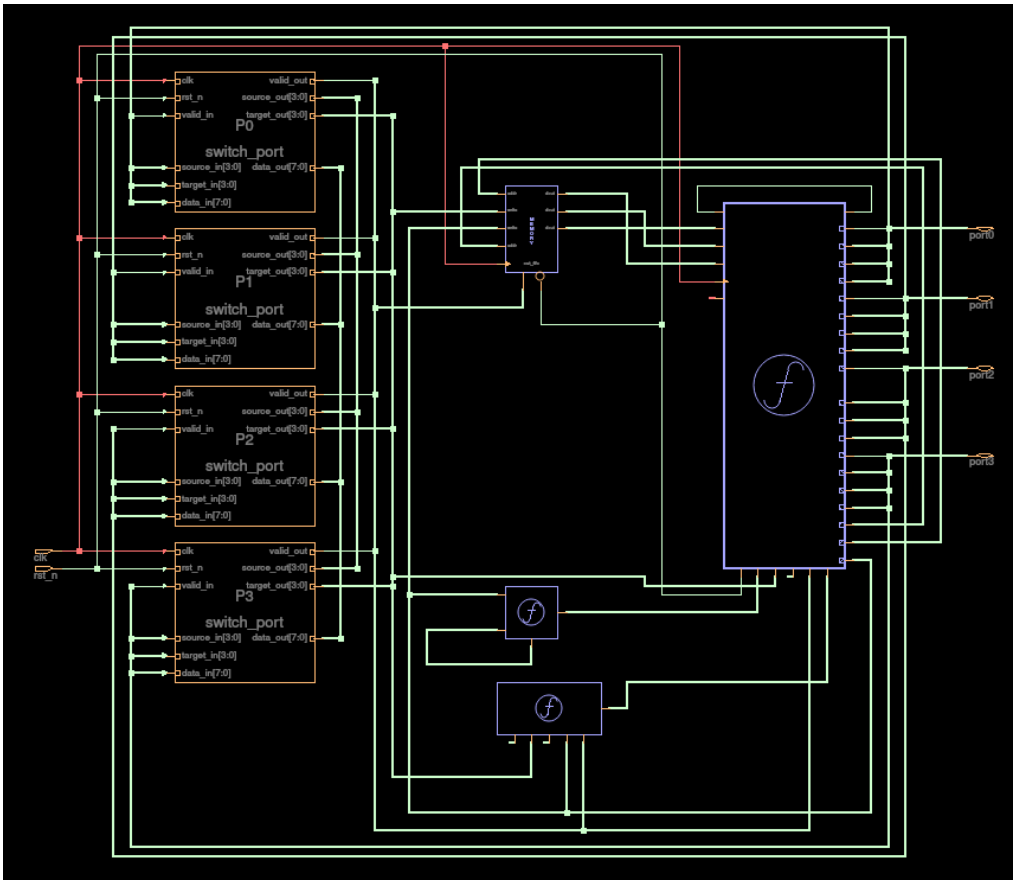
- **Frequency Optimization:** making sure the design achieves the target clock period with positive slack to support fast packet processing.
- **Area Minimization:** Reducing the physical silicon area by optimizing gate counts and resource utilization.
- **Power Efficiency:** Minimizing both static leakage and dynamic switching power by the implementation of clock gating.

The Synthesis Tool: fusion compiler. This tool was used to perform architectural synthesis, logic optimization, and technology mapping.

2- **Synthesis Process & Gate-Level Implementation :**

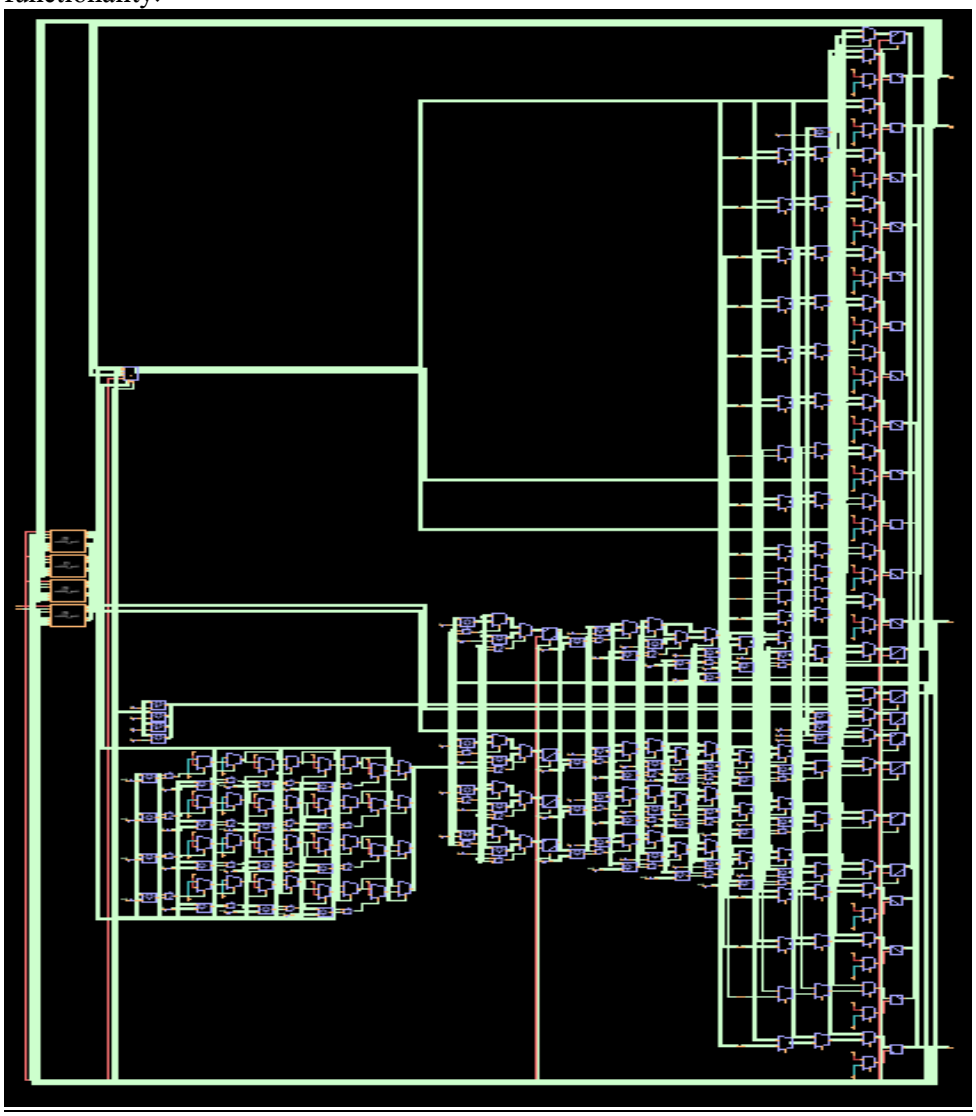
Modular non detailed view :

Before converting the design into gates, the tool provides a high level hierarchical view of the switch architecture.



Modular detailed view :

this view reveals the internal gate level logic (AND, OR, MUX) that implements the switch's functionality.



Synthesis script and tcl:

```
# remove any constraints that were applied from previous runs:
remove_sdc -design
set_disable_clock_gating_check [get_ports get_cells]

# Set the physical library:
set lib_name saed32hvt_ss0p75v125c

#set_current_design TOP

# Create clock object and set uncertainty
create_clock -period 10 [get_ports clk]
set_clock_uncertainty -setup 0.15 [get_clocks clk]
set_false_path -from [get_ports rst_n]

# Set constraints on input ports
#suppress_message UID-401
#suppress_message DCT-306


set_input_delay -max 0.5 -clock clk \
  [remove_from_collection [all_inputs] [get_ports {clk rst_n}]]


# Set constraints on output ports:
set_output_delay -max 0.2 -clock clk \
  [all_outputs]
# Handshake feedback is not timing-critical
set_false_path -from [get_ports in_valid[*]]  -to [get_ports in_ready[*]]
set_false_path -from [get_ports out_ready[*]] -to [get_ports out_valid[*]
```

Without clock gating :

```
#lappend search_path scripts design_data
set_host_options -max_cores 4
set TECH_FILE     "/data/synopsys/lib/saed32nm/ref/tech/saed32nm_1p9m.tf"
##########################################################################
### Physical Library Settings
##########################################################################
cd /a/home/cc/students/enginer/yazanatamna/project_leen_yazan
set lib_name switch_4port.dlib

if {![file exists $lib_name]} {
  create_lib -technology $TECH_FILE \
    -ref_libs { \
      /data/synopsys/lib/saed32nm/ref/CLIBs/saed32_hvt.ndm \
      /data/synopsys/lib/saed32nm/ref/CLIBs/saed32_lvt.ndm \
      /data/synopsys/lib/saed32nm/ref/CLIBs/saed32_rvt.ndm \
    } $lib_name
}

open_lib $lib_name

report_ref_libs
read_parasitic_tech -tlu /data/synopsys/lib/saed32nm/ref/tech/saed32nm_1p9m_Cmax.lv.tluplus -name Cmax
read_parasitic_tech -tlu /data/synopsys/lib/saed32nm/ref/tech/saed32nm_1p9m_Cmin.lv.tluplus -name Cmin
#
save_lib
analyze -format sverilog {
/a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_port.sv
/a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_4port.sv
/a/home/cc/students/enginer/yazanatamna/project_leen_yazan/port_if.sv
}
elaborate switch_4port
set_top_module switch_4port
# save hir
set_app_options -name  compile.flow.autoungroup  -value false
#start_gui
save_block -as switch_4port/elaborate

# mcmm_setup:
# Remove all MCMM related info
remove_corners   -all
remove_modes     -all
remove_scenarios -all
# Create Corners
create_corner Fast
create_corner Slow
#
## Set parasitics parameters
set_parasitics_parameters -early_spec_Cmin -late_spec  Cmin -corners {Fast}
set_parasitics_parameters -early_spec Cmax -late_spec  Cmax -corners {Slow}
#
## Create Mode
create_mode  FUNC
current_mode FUNC
#
## Create Scenarios
create_scenario -mode FUNC -corner Fast    -name FUNC_Fast
create_scenario -mode FUNC -corner Slow    -name FUNC_Slow
#

#sourse ConFiles/con431.con
current_scenario FUNC_Fast
source /a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_4port.sdc
current_scenario FUNC_Slow
source /a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_4port.sdc

set_auto_floorplan_constraints -core_utilization 0.7 -side_ratio {1 1} -core_offset 2

set_clock_gating_options -minimum_bitwidth 1000000 -max_fanout 1000000

set_clock_gating_enable -exclude [all_registers]

compile_fusion -to logic_opto
#create_placement
#legalize_placement
##Power
compile_fusion -to final_opto

save_block -as switch_4port/final_opto

#set_attribute [get_layers {M1 M3 M5 M7 M9}]   routing_direction vertical
#set_attribute [get_layers {M2 M4 M6 M8 MRDL}] routing_direction horizontal

#source create_pg_network.tcl
#clock_opt
#route_opt
#save_block -as ALU/route_opt
```

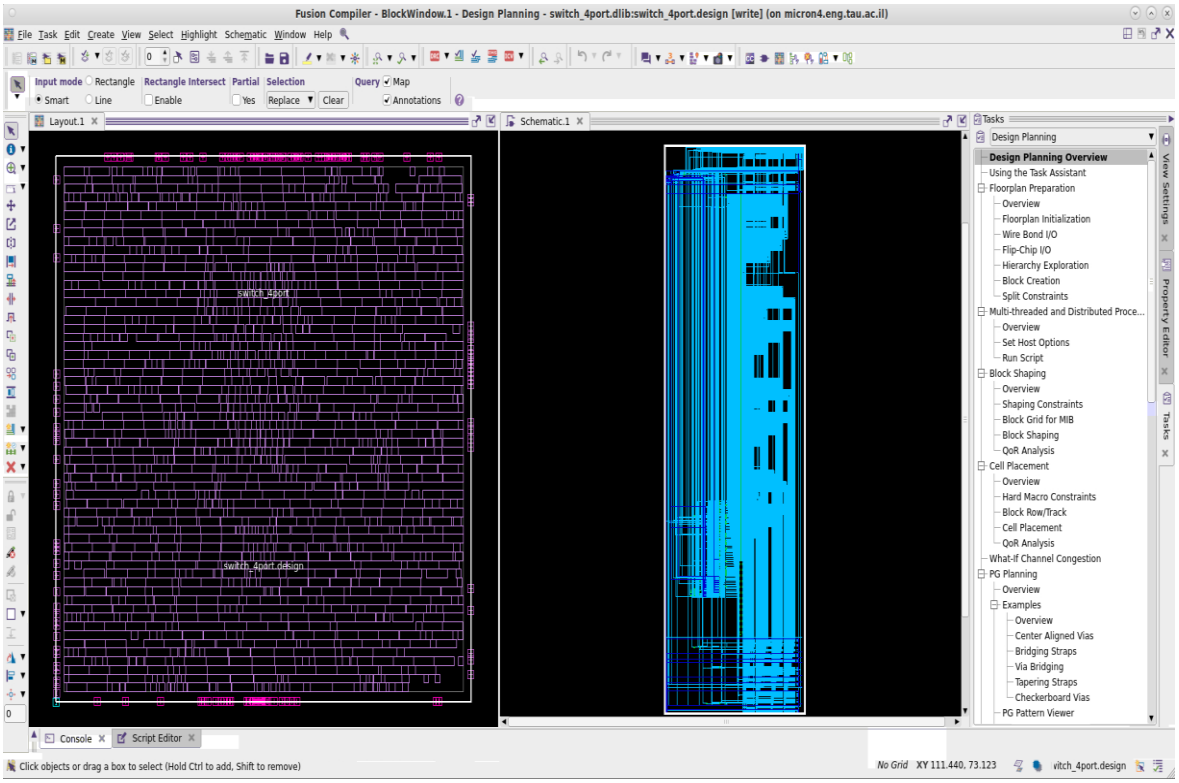
With clock gating:

```
#lappend search_path scripts design_data
set_host_options -max_cores 4
set TECH_FILE     "/data/synopsys/lib/saed32nm/ref/tech/saed32nm_1p9m.tf"
##########################################################################
### Physical Library Settings
##########################################################################
cd /a/home/cc/students/enginer/yazanatamna/project_leen_yazan
set lib_name switch_4port.dlib

if {![file exists $lib_name]} {
  create_lib -technology $TECH_FILE \
    -ref_libs { \
      /data/synopsys/lib/saed32nm/ref/CLIBs/saed32_hvt.ndm \
      /data/synopsys/lib/saed32nm/ref/CLIBs/saed32_lvt.ndm \
      /data/synopsys/lib/saed32nm/ref/CLIBs/saed32_rvt.ndm \
    } $lib_name
}

open_lib $lib_name

report_ref_libs
read_parasitic_tech -tlu /data/synopsys/lib/saed32nm/ref/tech/saed32nm_1p9m_Cmax.lv.tluplus -name Cmax
read_parasitic_tech -tlu /data/synopsys/lib/saed32nm/ref/tech/saed32nm_1p9m_Cmin.lv.tluplus -name Cmin
#
save_lib
analyze -format sverilog {
/a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_port.sv
/a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_4port.sv
/a/home/cc/students/enginer/yazanatamna/project_leen_yazan/port_if.sv
}
elaborate switch_4port
set_top_module switch_4port
# save hir
set_app_options -name  compile.flow.autoungroup  -value false
#start_gui
save_block -as switch_4port/elaborate

# mcmm_setup:
# Remove all MCMM related info
remove_corners   -all
remove_modes     -all
remove_scenarios -all
# Create Corners
create_corner Fast
create_corner Slow
#
## Set parasitics parameters
set_parasitics_parameters -early_spec Cmin -late_spec  Cmin -corners {Fast}
```

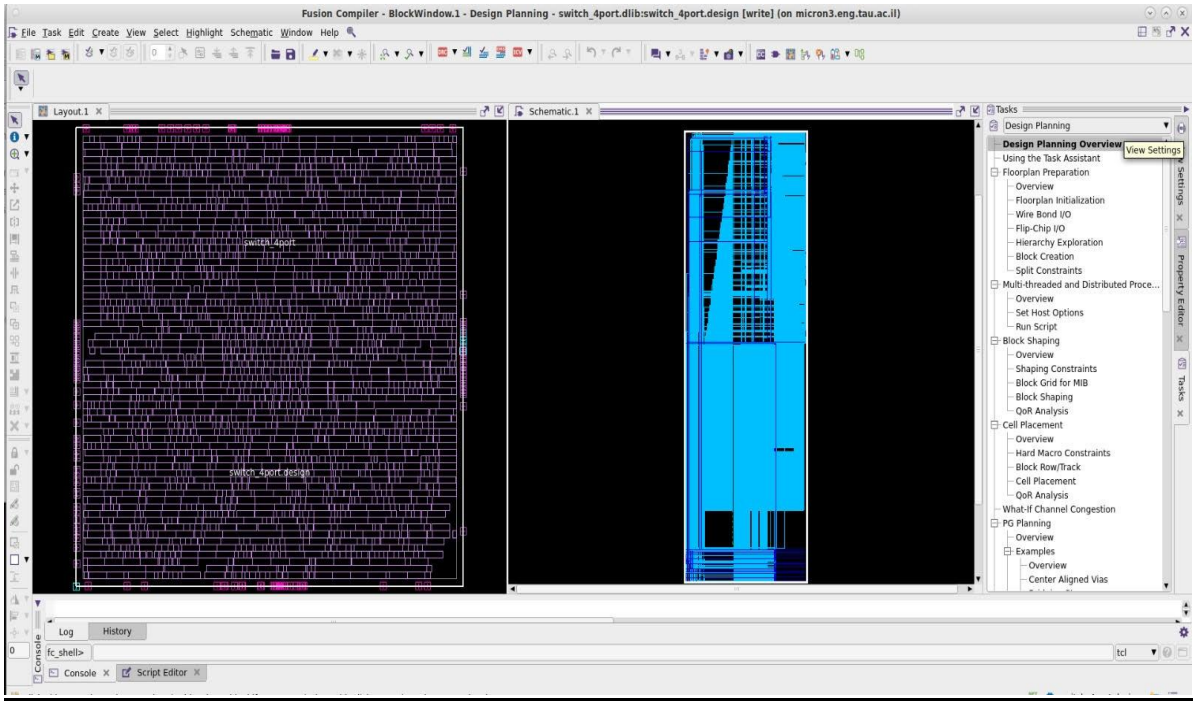
```
set_parasitics_parameters -early_spec Cmax -late_spec  Cmax -corners {Slow}
#
## Create Mode
create_mode  FUNC
current_mode FUNC
#
## Create Scenarios
create_scenario -mode FUNC -corner Fast    -name FUNC_Fast
create_scenario -mode FUNC -corner Slow    -name FUNC_Slow
#

#sourse ConFiles/con431.con
current_scenario FUNC_Fast
source /a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_4port.sdc
current_scenario FUNC_Slow
source /a/home/cc/students/enginer/yazanatamna/project_leen_yazan/switch_4port.sdc

set_auto_floorplan_constraints -core_utilization 0.7 -side_ratio {1 1} -core_offset 2


 compile_fusion -to logic_opto
 #create_placement
 #legalize_placement
 ##Power
 compile_fusion -to final_opto

 save_block -as switch_4port/final_opto

 #set_attribute [get_layers {M1 M3 M5 M7 M9}]   routing_direction vertical
 #set_attribute [get_layers {M2 M4 M6 M8 MRDL}] routing_direction horizontal

 #source create_pg_network.tcl
 #clock_opt
 #route_opt
 #save_block -as ALU/route_opt
```

Synthesis screenshot:

With clock gating:



Without clock gating :



3-clock gating report :

a. Clock Gating Effectiveness:

The synthesis process utilized Integrated Clock Gating (ICG) to reduce dynamic power by automatically disabling the clock signal for registers during idle periods.

- Total Registers: 596
- Gated Registers: 584 (97.99%)
- Ungated Registers: 12 (2.01%)
- Clock Gating Elements (ICGs): 48

```
****************************************
Report : clock_gating
Design : switch_4port
Version: V-2023.12-SP3
Date   : Fri Jan 16 13:54:23 2026
****************************************
```

Clock Gating Summary

Number of Clock gating elements	48	
Number of Tool-Inserted Clock gates	48	(100.00%)
Number of Pre-Existing Clock gates	0	(0.00%)
Number of Gated registers	584	(97.99%)
Number of Tool-Inserted Gated registers	584	(97.99%)
Number of Pre-Existing Gated registers	0	(0.00%)
Number of Ungated registers	12	(2.01%)
Total number of registers	596	
Max number of Clock Gate Levels	1	

4-Results: with clock gating:

Time report analysis:

a. Path:

- Critical Path Start point: P3/fifo_valid_reg (Port 3 FIFO control logic).
- Critical Path End point: clock_gate_out_fifo_reg_25/EN (The Enable pin of the Integrated Clock Gating (ICG) cell).
- Path Type: Max (Setup check).

b.Timing Results and Calculations

- Target Clock Period: 10ns .
- Data Arrival Time: 2.11ns This is the time it takes for the signal to travel through the logic gates to reaching the clock gate enable pin.
- Data Required Time: 9.05ns.

- Worst Negative Slack (WNS): 6.94 (MET). Since the slack is positive, the design easily meets the 10ns target.

c. Max Frequency:

$$\frac{1}{period - slack} = \frac{1}{(10 - 6.94)ns} = 326.8MHz$$

```
****************************************
Report : timing
         -path_type full
         -delay_type max
         -max_paths 1
         -report_by design
Design : switch_4port
Version: V-2023.12-SP3
Date   : Fri Jan 16 14:15:17 2026
****************************************

  Startpoint: P3/fifo_valid_reg (rising edge-triggered flip-flop clocked by c
  Endpoint: clock_gate_out_fifo_reg_25 (rising clock gating-check end-point c
  Mode: FUNC
  Corner: Slow
  Scenario: FUNC_Slow
  Path Group: clk
  Path Type: max

  Point                                                  Incr       Path
  -----------------------------------------------------------------------
  clock clk (rise edge)                                  0.00       0.00
  clock network delay (ideal)                            0.00       0.00

  P3/fifo_valid_reg/CLK (SDFFARX1_HVT)                   0.00       0.00 r
  P3/fifo_valid_reg/Q (SDFFARX1_HVT)                     0.56       0.56 f
  P3/ctmi_98/Y (AND3X1_RVT)                              0.25       0.81 f
  ctmi_3929/Y (AO22X1_RVT)                               0.22       1.04 f
  ctmi_3928/Y (AO21X1_RVT)                               0.15       1.19 f
  HFSBUF_696_647/Y (NBUFFX2_RVT)                         0.17       1.36 f
  ctmi_3927/Y (AO21X1_RVT)                               0.15       1.52 f
  ctmi_3932/Y (NAND2X0_RVT)                              0.15       1.67 r
  phfnr_buf_625/Y (INVX0_RVT)                            0.15       1.82 f
  ctmi_3942/Y (AND4X1_RVT)                               0.28       2.11 f
  clock_gate_out_fifo_reg_25/EN (CGLPPRX2_HVT)           0.00       2.11 f
  data arrival time                                                 2.11

  clock clk (rise edge)                                 10.00      10.00
  clock network delay (ideal)                           -0.34       9.66
  clock_gate_out_fifo_reg_25/CLK (CGLPPRX2_HVT)          0.00       9.66 r
  clock uncertainty                                     -0.15       9.51
  library setup time                                    -0.46       9.05
  data required time                                                9.05
  -----------------------------------------------------------------------
  data required time                                                9.05
  data arrival time                                                -2.11
  -----------------------------------------------------------------------
  slack (MET)                                                       6.94
```

Area report analysis:

a. Cell Summary: The design consists of 1,296 total cells:

- Sequential Cells (644): These represent the registers (flip-flops) used in the FIFOs, FSM state registers, and the crossbar data registers.
- Combinational Cells (652): These implement the logic that helped us implement the functionality of our switch.
- Buffers/Inverters (106): These are used for signal drive strength and timing fix-ups across the four ports.

b. Area Distribution:

- Non-combinational Area: 5494.59 (almost 76% of total area).

- Combinational Area: 1704.04 (almost 24% of total area).

c. density anaylsis: The high ratio of non combinational area suggests that the design's footprint is mainly driven by the data storage (FIFOs) for the four ports. While clock gating adds a small amount of combinational logic (enable gates), it is so low compared to the area of the registers themselves.

d. 0 Macros/black boxes the entire switch has been synthesized using standard cells.

```
****************************************
Report : area
Design : switch_4port
Version: V-2023.12-SP3
Date   : Fri Jan 16 14:22:10 2026
****************************************

Number of ports:                          304
Number of nets:                          1564
Number of cells:                         1296
Number of combinational cells:            652
Number of sequential cells:               644
Number of macros/black boxes:               0
Number of buf/inv:                        106
Number of references:                      35

Combinational area:                  1704.04
Buf/Inv area:                         214.75
Noncombinational area:               5494.59
Macro/Black Box area:                   0.00

Total cell area:                     7198.63
```

Power report analysis:

a. Total Power Breakdown: the total power is the sum of Dynamic power (switching activity) and Leakage power (static).:

- Total Dynamic Power: 69.3mW
    - Internal Power (90.1%): Power consumed inside the cells during switching.
    - Switching Power (9.9%): Power consumed charging/discharging the capacitive load of the nets.
- Cell Leakage Power: 135mW
- Total Power Consumption: 204.3mW

b. power by groups:

1. Register Power (66.2%):

The registers are the largest power consumers in the design. This is expected for a packet switch for two reasons: 1.data storage: The four ports contain FIFOs that must hold packet data (64-bit payloads) while waiting for arbitration. 2.leakage: registers contribute the most to leakage power since registers are built from many transistors that stay on to maintain state, they leak significantly more than simple gates.

2. Combinational Power (17.3%) : Because this logic only switches when a packet is being processed, its Switching Power is relatively low 3.56mW

3. Clock Network Power (16.6%):

The clock network is usually the most expensive part of a chip because it toggles every single cycle. However, in this design, it is kept to only 16.6% due to the implementation of cg ,By disabling the clock for ports that are not receiving or transmitting data, we prevent thousands of internal register transitions, effectively "freezing" the power consumption of idle sub-modules.

```
Warning: Power table extrapolation (extrapolation mode) for port GCLK on cell P0/clock_gate_fifo_data_reg for parameter Cout. Lowest table
value = 0.000100, highest table value = 0.016000, value = 0.016217 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port D on cell out_fifo_reg[2][1][data][6] for parameter Tinp. Lowest table
value = inf, highest table value = inf, value = 0.160217 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port CLK on cell out_fifo_reg[2][1][data][6] for parameter Tinp. Lowest table
value = 0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port Q on cell out_fifo_reg[2][1][data][6] for parameter Tinp. Lowest table
value = 0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell out_fifo_reg[2][1][data][6] for parameter Tinp. Lowest table
value = 0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell out_fifo_reg[2][1][data][6] for parameter Cout. Lowest table
value = 0.000100, highest table value = 0.008000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port D on cell out_fifo_reg[2][1][data][5] for parameter Tinp. Lowest table
value = inf, highest table value = inf, value = 0.161228 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port CLK on cell out_fifo_reg[2][1][data][5] for parameter Tinp. Lowest table
value = 0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Note - message 'POW-046' limit (10) exceeded. Remainder will be suppressed.

  Cell Internal Power    = 6.24e+07 pW ( 90.1%)
  Net Switching Power    = 6.86e+06 pW (  9.9%)
Total Dynamic Power      = 6.93e+07 pW (100.0%)

Cell Leakage Power       = 1.35e+08 pW


  Attributes
  ----------
      u  -  User defined power group
      i  -  Includes clock pin internal power

Power Group         Internal Power       Switching Power         Leakage Power           Total Power    (   %  )    Attrs
--------------------------------------------------------------------------------------------------------------------------
io_pad                   0.00e+00             0.00e+00             0.00e+00             0.00e+00    (  0.0%)
memory                   0.00e+00             0.00e+00             0.00e+00             0.00e+00    (  0.0%)
black_box                0.00e+00             0.00e+00             0.00e+00             0.00e+00    (  0.0%)
clock_network            2.93e+07             2.31e+06             2.28e+06             3.39e+07    ( 16.6%)       i
register                 3.00e+07             9.92e+05             1.04e+08             1.35e+08    ( 66.2%)
sequential               0.00e+00             0.00e+00             0.00e+00             0.00e+00    (  0.0%)
combinational            3.06e+06             3.56e+06             2.87e+07             3.53e+07    ( 17.3%)
--------------------------------------------------------------------------------------------------------------------------
Total                    6.24e+07 pW          6.86e+06 pW          1.35e+08 pW          2.05e+08 pW
```

Runtime Performance :

The efficiency of the synthesis tool flow was evaluated to understand the computational cost of the optimized design.

Synthesis Session Summary:

We got the following values:

- Total Session Runtime (Wall-clock): 00:03:14 (194 seconds)
- Total CPU Time: 00:02:38 (158 seconds)
- Peak Memory Utilization: 1474 MB

Efficiency Analysis

For the runtime summary we can conclude highly efficient synthesis process for the four-port switch architecture:

- Metric Distinction: The Session Runtime (03:14) represents the total time, including environment setup, logic elaboration, and report generation. The CPU Time (02:38) reflects the

actual processing time consumed by the compute engine. The ratio between these values indicates low overhead and high tool throughput.

- Optimization Effort: For a design of approximately 1,300 cells and 600 registers, a 3 minute runtime is considered optimal.

- With a memory footprint of 1474 MB, the synthesis session was lightweight enough to run on standard compute nodes without hitting memory swap limits.

5-Results: without clock gating:

Time analysis:

a. **Path Definition:**

- Critical Path Startpoint: P3/fifo_valid_reg (Port 3 FIFO control logic).
- Critical Path Endpoint: out_fifo_reg[2][6][data][6] (A data register in the Output FIFO).
- Path Type: Max (Setup check).
- Endpoint Explanation: Without clock gating logic, the critical path now transitions through the main data switching fabric directly to the output storage registers, rather than ending at a clock-gating check.

b. **Timing Results and Calculations:**

- Target Clock Period: 10.00 ns
- Data Arrival Time: 6.10 ns.
- Data Required Time: 9.58 ns
- Worst Negative Slack (WNS): +3.49 ns (MET). The design meets the 10ns target, though with a smaller margin compared to the gated version.

c. **Maximum frequency:**

$$\frac{\mathbf{1}}{\boldsymbol{period - slack}} = \frac{\mathbf{1}}{(\mathbf{10 - 3.49})\boldsymbol{ns}} = \mathbf{153.6}\boldsymbol{MHz}$$

```
****************************************
Report : timing
         -path_type full
         -delay_type max
         -max_paths 1
         -report_by design
Design : switch_4port
Version: V-2023.12-SP3
Date   : Mon Jan 19 14:41:50 2026
****************************************

  Startpoint: P3/fifo_valid_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: out_fifo_reg[2][6][data][6] (rising edge-triggered flip-flop clocked by clk)
  Mode: FUNC
  Corner: Slow
  Scenario: FUNC_Slow
  Path Group: clk
  Path Type: max

  Point                                                   Incr       Path
  ---------------------------------------------------------------------------
  clock clk (rise edge)                                   0.00       0.00
  clock network delay (ideal)                             0.00       0.00

  P3/fifo_valid_reg/CLK (SDFFARX1_HVT)                    0.00       0.00 r
  P3/fifo_valid_reg/Q (SDFFARX1_HVT)                      0.62       0.62 r
  P3/ctmi_230/Y (AND3X2_HVT)                              1.86       2.48 r
  ctmi_6137/Y (AO22X1_RVT)                                0.34       2.81 r
  ctmi_6136/Y (AO21X1_RVT)                                0.14       2.95 r
  ctmi_6135/Y (AO21X1_RVT)                                0.16       3.12 r
  ctmi_6841/Y (AND3X1_RVT)                                0.19       3.31 r
  ctmi_6981/Y (NAND3X4_HVT)                               1.90       5.21 f
  HFSINV_492_870/Y (INVX1_HVT)                            0.47       5.68 r
  HFSBUF_444_869/Y (NBUFFX2_RVT)                          0.25       5.93 r
  ctmi_6990/Y (AO22X1_RVT)                                0.17       6.10 r
  out_fifo_reg[2][6][data][6]/D (SDFFX1_RVT)              0.00       6.10 r
  data arrival time                                                  6.10

  clock clk (rise edge)                                  10.00      10.00
  clock network delay (ideal)                             0.00      10.00
  out_fifo_reg[2][6][data][6]/CLK (SDFFX1_RVT)            0.00      10.00 r
  clock uncertainty                                      -0.15       9.85
  library setup time                                     -0.27       9.58
  data required time                                                 9.58
  ---------------------------------------------------------------------------
  data required time                                                 9.58
  data arrival time                                                 -6.10
  ---------------------------------------------------------------------------
  slack (MET)                                                        3.49
```

Area report analysis:

a. Cell Summary: The design consists of 1,873 total cells:

- Sequential Cells (596)
- Combinational Cells (1,277)
- Buffers/Inverters (181)

b. Area Distribution:

- Non-combinational Area: 5216.56 (almost 61.3% of total area).
- Combinational Area: 3286.84 (almost 38.7% of total area).

c. Density Analysis: The area distribution shows a significantly higher proportion of combinational logic (38.7%) compared to the gated version (24%). This indicates that without the specific constraints and optimizations triggered by clock gating, the synthesis tool utilized more combinational standard cells to map the design's logic. However, data storage remains a major driver of the total footprint.

d. 0 Macros/Black Boxes: the entire switch has been synthesized using standard cells.

```
****************************************
Report : area
Design : switch_4port
Version: V-2023.12-SP3
Date   : Mon Jan 19 14:41:09 2026
****************************************

Number of ports:                          303
Number of nets:                          2144
Number of cells:                         1873
Number of combinational cells:           1277
Number of sequential cells:               596
Number of macros/black boxes:               0
Number of buf/inv:                        181
Number of references:                      46

Combinational area:                   3286.84
Buf/Inv area:                          339.79
Noncombinational area:                5216.56
Macro/Black Box area:                    0.00

Total cell area:                      8503.40
```

Power report analysis:

a.Total Power Breakdown:

- Total Dynamic Power: 250.0 mW
    1- Internal Power (97.6%)
    2- Switching Power (2.4%)
- Cell Leakage Power: 159.0 mW
- Total Power Consumption: 409.0 mW

b. Power by Group:

1. Clock Network Power (50.8%):

   Now, the clock network is the largest power consumer which is 208.0 mW. Without clock gating, the clock signal toggles every cycle for all 596 registers, regardless of whether the switch is processing a packet or is idle. Therefore the clock network consumes more than half of the total power budget.

2. Register Power (32.9%):

   The registers consume 134.0 mW. While they remain critical for data storage in the FIFOs, their relative contribution to the total power is lower than in the gated version because the unoptimized clock network now dominates the power profile. Leakage remains a significant factor for this group due to the high transistor count required to maintain state.

3. Combinational Power (16.3%):

   The combinational logic consumes 66.4 mW. Similar to the gated version, this power is consumed by the arbitration and routing logic. Its switching power remains relatively low (5.17 mW) because these gates only toggle during active packet transitions.

```
0.016000, highest table value = 1.024000, value = 0.000648 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port Q on cell P0/lat_cnt_reg[0] for parameter Tinp. Lowest table value =
0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell P0/lat_cnt_reg[0] for parameter Tinp. Lowest table value =
0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port QN on cell P0/lat_cnt_reg[0] for parameter Cout. Lowest table value =
0.000100, highest table value = 0.008000, value = 0.000000 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port A4 on cell P0/ctmi_243 for parameter Tinp. Lowest table value = 0.016000,
highest table value = 1.024000, value = 0.000019 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port Y on cell P0/ctmi_243 for parameter Tinp. Lowest table value = 0.016000,
highest table value = 1.024000, value = 0.000019 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port D on cell P0/fifo_source_reg[2] for parameter Tinp. Lowest table value =
inf, highest table value = inf, value = 0.063992 (POW-046)
Warning: Power table extrapolation (extrapolation mode) for port CLK on cell P0/fifo_source_reg[2] for parameter Tinp. Lowest table value
= 0.016000, highest table value = 1.024000, value = 0.000000 (POW-046)
Note - message 'POW-046' limit (10) exceeded. Remainder will be suppressed.

  Cell Internal Power    = 2.44e+08 pW ( 97.5%)
  Net Switching Power    = 6.34e+06 pW (  2.5%)
Total Dynamic Power      = 2.50e+08 pW (100.0%)

Cell Leakage Power       = 1.59e+08 pW


  Attributes
  ----------
      u  -  User defined power group
      i  -  Includes clock pin internal power

Power Group         Internal Power       Switching Power         Leakage Power           Total Power    (   %   )    Attrs
--------------------------------------------------------------------------------------------------------------------------
io_pad                  0.00e+00             0.00e+00              0.00e+00              0.00e+00    (  0.0%)
memory                  0.00e+00             0.00e+00              0.00e+00              0.00e+00    (  0.0%)
black_box               0.00e+00             0.00e+00              0.00e+00              0.00e+00    (  0.0%)
clock_network           2.08e+08             0.00e+00              0.00e+00              2.08e+08    ( 50.8%)       i
register                3.16e+07             1.16e+06              1.02e+08              1.34e+08    ( 32.9%)
sequential              0.00e+00             0.00e+00              0.00e+00              0.00e+00    (  0.0%)
combinational           4.36e+06             5.17e+06              5.69e+07              6.64e+07    ( 16.3%)
--------------------------------------------------------------------------------------------------------------------------
Total                   2.44e+08 pW          6.34e+06 pW           1.59e+08 pW           4.09e+08 pW
```

Runtime performance:

For this run, the tool optimized the architecture without the additional computational overhead required for clock gating insertion.

a. Synthesis Session Summary

We got these values from the Fusion Compiler session log:

- Total Session Runtime (Wall-clock): 00:03:10 (190 seconds)
- Total CPU Time: 00:02:48 (168 seconds)
- Peak Memory Utilization: 1491 MB

b. Efficiency Analysis

The runtime results shows highly efficient synthesis process:

- Metric Distinction: The Session Runtime (03:10) represents the total time. The CPU Time (02:48) shows the processing resources consumed by the engine. The 190-second total runtime is slightly faster than the gated version (194s), as expected when power-optimization passes are reduced.
- Optimization Effort: For a design of approximately 1,873 cells and 596 registers, a 3-minute runtime is considered optimal for reaching a stable gate-level implementation.
- Resource Utilization: With a memory footprint of 1491 MB, the session remained highly efficient without exceeding memory swap limits.

6- **Comparison analysis:**

The table below compares between multiple factors that are affected by clock gating.

aspect	With clock gating	Without clock gating	conclusion
Max Frequency	326.8 MHz,	153.6 MHz	Improvement of +112%
Worst negative slack	6.94 ns	3.49 ns	3.45 ns diffrence
Total cell area	7198 units	8503 units	-15.3% reduction
Total cell count	1296 cells	1873 cells	-30.8% reduction
Total dynamic power	69.3mW	250 mW	-72% reduction
Total leakage power	135 mW	159 mW	-15 % (improvement)
Total power	409 mW	204.3 mW	-50 % reduction
Clock gating efficiency	Optimized 97.995% of registers	Optimized 0%	Completely changed (trivial)
Synthesis runtime	194 sec	190 sec	+2.1%

Summary of Analysis

- Power Efficiency: the most significal change was the 50% reduction in the total power .without gating, the clock network consumes more than half of the power budget (50.8%) because every register toggles every cycle. However with gating, this is reduced to 16.6%.
- Performance: the clock gating led to a big increase in the max frequency therefore we can conclude that the optimization passes in fusion gave us a more efficient timing mapped netlist
- Area: optimized version is 15.3% smaller. The tool successfully reduced the combinational cell count from 1,277 to 652 cells during the gated-synthesis flow.
- Runtime: Achieving these great results only added 4 seconds to the total synthesis session time.

Summary of the project :

In this project we took a four-port packet switch from a high level concept to a gate level netlist. We moved between the 3 stages : design verification and synthesis basically transitioning from a code to optimized gates . by adding clock gating we reduced the power by 50% successfully and at the same time we succeeded to achieve a frequency of 326.8MHz.