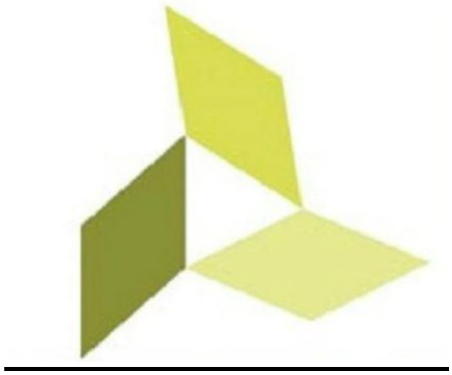
PROJECT WORK

"Verilog Implementation of a Voting Machine Circuit Designs on FPGA using Vivado"

Topic: Voting Machine System Design using Verilog



Tatchemo Guiafaing Ronald VR512344

28th April 2024

1st Year Master's Student in Computer Engineering for Robotic's and Smart Industry

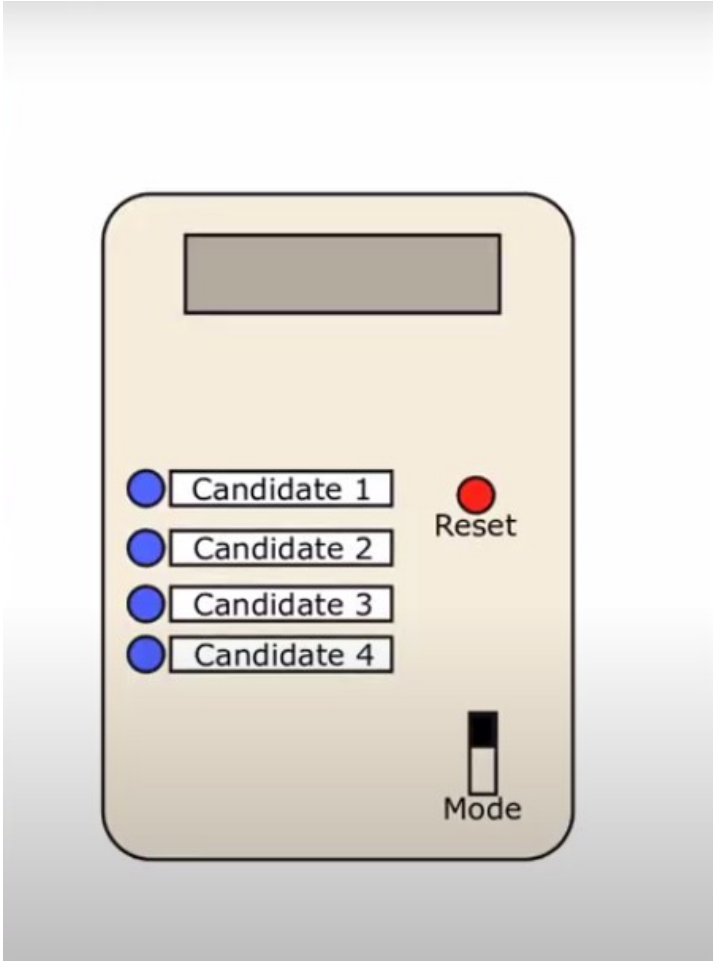
University of Verona (UNVR)

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Professor Luigi Capogrosso and Professor Michele Lora for their exemplary guidance, monitoring and constant encouragement throughout the course of this project.

PROBLEM STATEMENT

The aim of the project is to design a Voting Machine System for 4 Different Candidate thereby giving to any User the possibility to vote between one of those 4 Candidates. Users can be able to press one of the four buttons to vote for a candidate, and the machine stores the vote accordingly. The System also have a Display Percentage Option thereby given possibility for Pressing any of the four buttons displays the percentage (rounded to an integer) of votes received by the corresponding candidate.



As System implementation is Made up of 4 design Schema

1-VotingMachine.v

it has inputs for **clock, reset**, **mode**, and **buttons** for each candidate, as well as outputs for vote counts and LEDs. Vote Counting Process: On each clock cycle, it increments the vote counts for candidates based on button presses when in vote collection mode. Reset Functionality: When reset is asserted, all vote counts are reset to zero.

Output Assignment: The module assigns the current vote counts to corresponding output ports.

LED Output: There's a port for LED output, although it's not used in this portion of the code.

Other Modules: The code doesn't instantiate any other modules, but it's indicated for potential expansion.

Overall, this module provides the basic functionality of a voting machine, where button presses increment vote counts for candidates, and the counts can be read externally.

2-ButtonControl.v

It is designed to de-bounce a button input and generate a valid vote signal. Here's a summary:

- **Inputs:** Clock, reset, button.
- **Outputs:** Valid_vote.
- **Internal Signal:** Counter (2-bit register).
- **Parameter:** COUNTER_LIMIT (set to 100,000,000).
  Functionality:
- On each clock cycle, if reset is asserted, the counter is reset.
- If the button is pressed and the counter has not reached its limit, the counter increments, effectively de-bouncing the button.
- When the button is released, the counter resets.
- Another process checks if the counter reaches the limit, indicating a valid vote, and sets valid_vote accordingly.
- **Purpose:** Ensures that only sustained button presses beyond a certain duration are considered as valid votes, filtering out noise or bouncing from the button.
  This module effectively filters out noise or bouncing from the button signal and generates a single, valid vote signal once the button has been held down for a certain duration, ensuring accurate voting.

3-ModeControl.v

This Verilog module, controls the operation of the voting machine based on different modes. Here's a summary of its functionality:

Module Description:

- Inputs: clock, reset, mode, valid_vote_casted, candidate vote counts, and buttons for each candidate.
- Outputs: LEDs.
- Internal Signals:
- sync_reset: Synchronized reset signal.
- leds_internal: Internal signal for LED logic.
- candidate_votes: Array to hold vote counts for each candidate.
- counter: Counter for mode switching and vote counting.
- total_votes: Total number of votes received.
- mode_previous: Previous mode value for detecting mode changes.
-

Initialization:

- On reset, initializes counters and vote counts.

Mode Switching:

- Tracks mode changes and resets vote counts accordingly.

LED Display Logic:

- Displays different patterns based on the current mode and button presses.

Vote Counting Logic:

- Increments total votes and candidate votes based on valid votes received.

Output Assignment:

- Assigns the internal LED signal to the output port, synchronized with the clock.

Overall, this module manages the voting machine's operation, including mode switching, vote counting, LED display logic, and output control based on button presses and mode changes.

4-voteLogger.v

It is Principally responsible for logging valid votes received for each candidate. Here's a summary of its functionality:

Module Description:

- Inputs: clock, reset, mode, and signals indicating valid votes for each candidate.
- Outputs: Registers to store the received votes for each candidate.

Vote Logging Logic:

- On each clock cycle, if reset is asserted, the stored vote counts for all candidates are reset to zero.
- If the mode is set to 0 (indicating vote collection mode):
- The module increments the stored vote count for each candidate if a valid vote signal is received for that candidate.

Initialization:

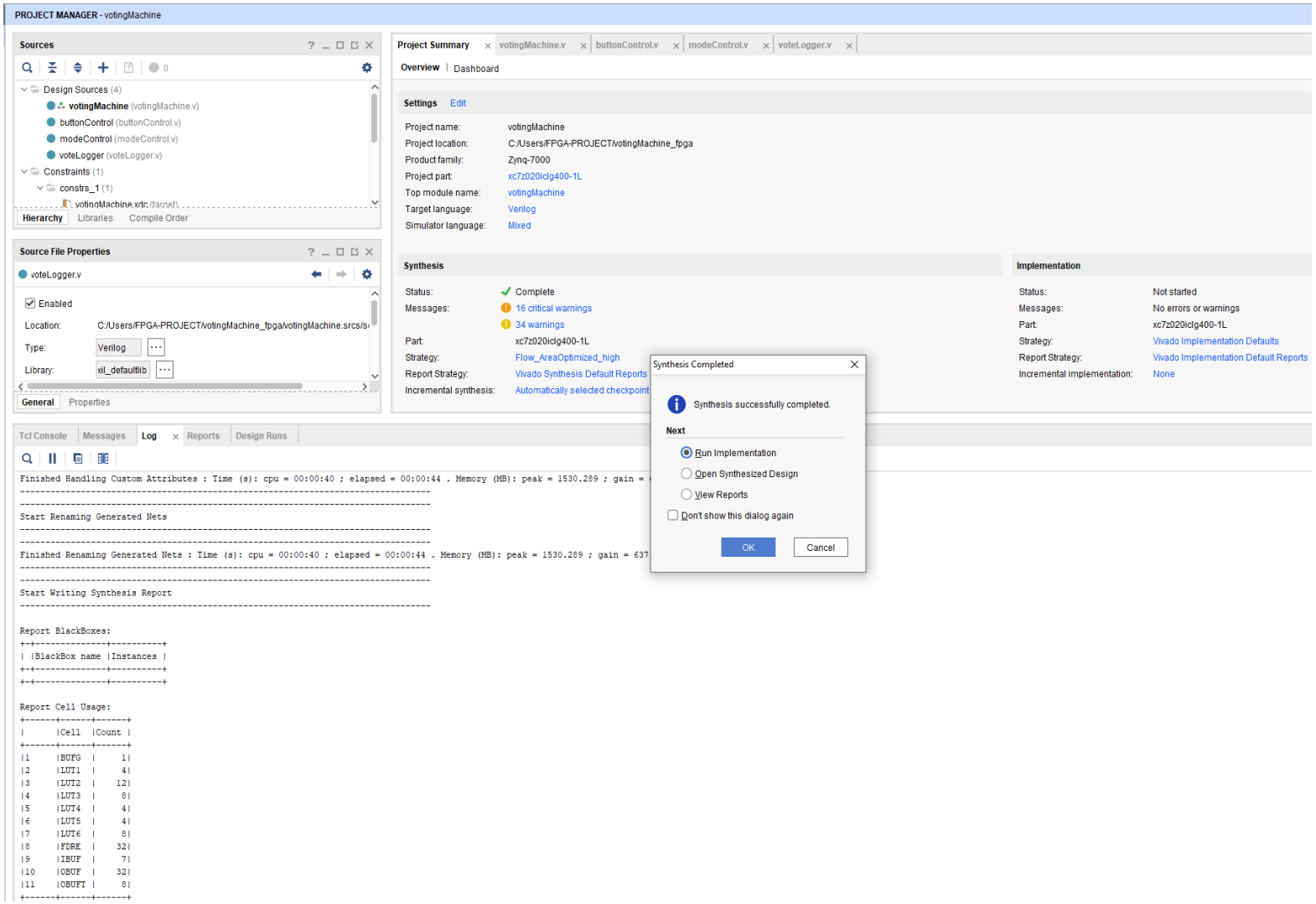
- On reset, all stored vote counts are reset to zero.

Output Assignment:

- The module assigns the updated vote counts to the corresponding output registers.

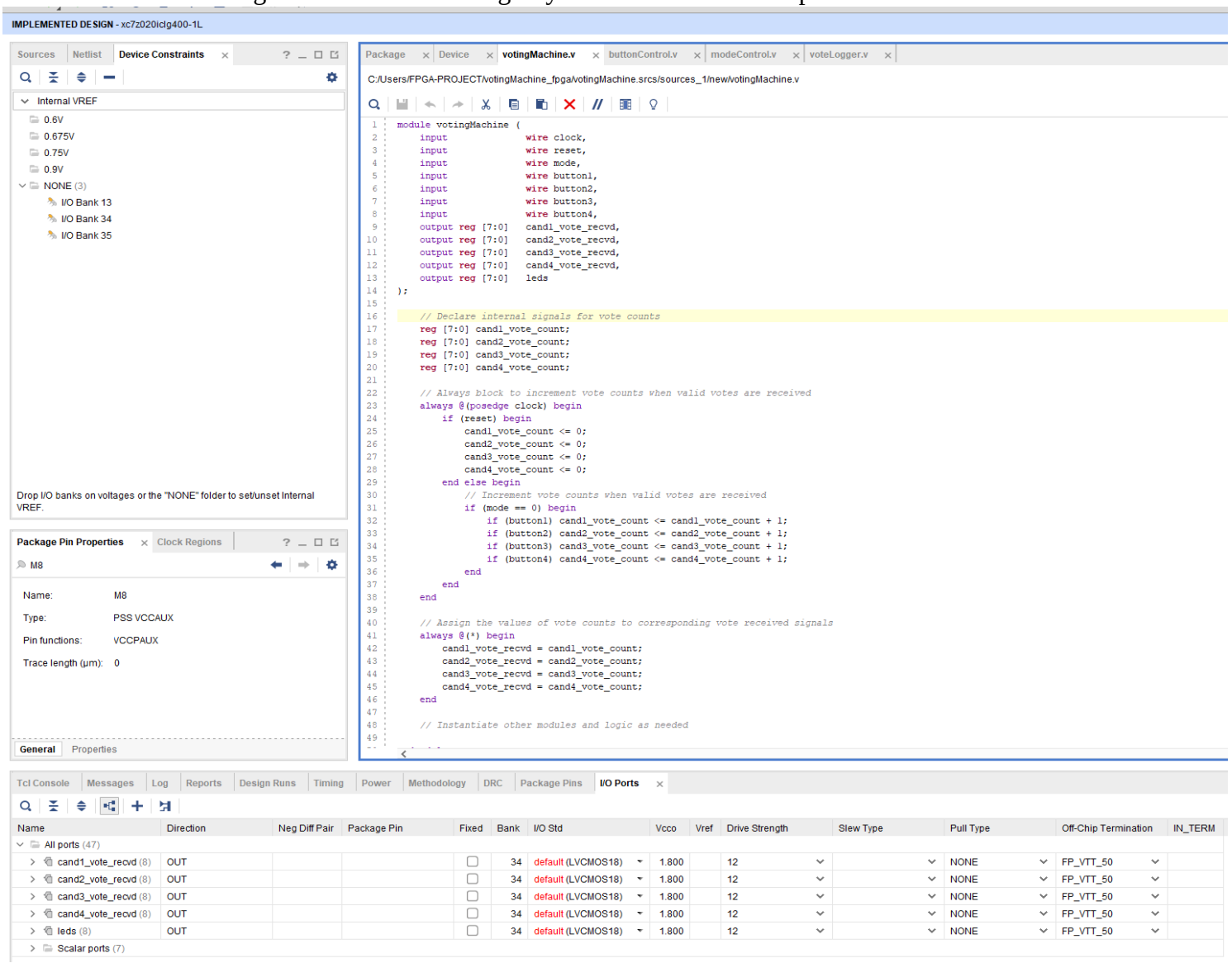
Overall, this module acts as a simple vote logger, incrementing the vote count for each candidate when a valid vote signal is received, and providing the current count as output.

Those 4 Synthesis compilation is Visible as below.

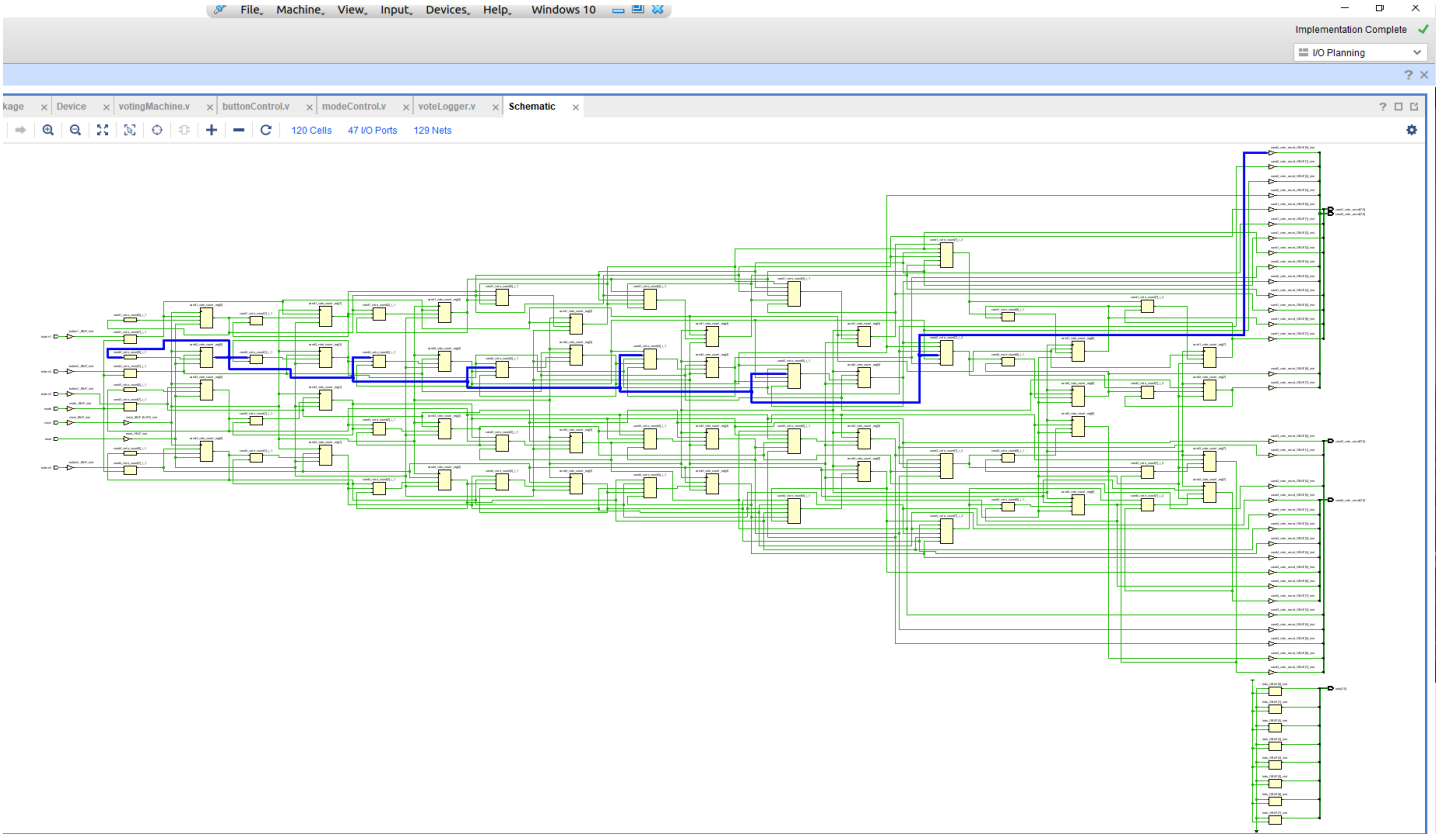


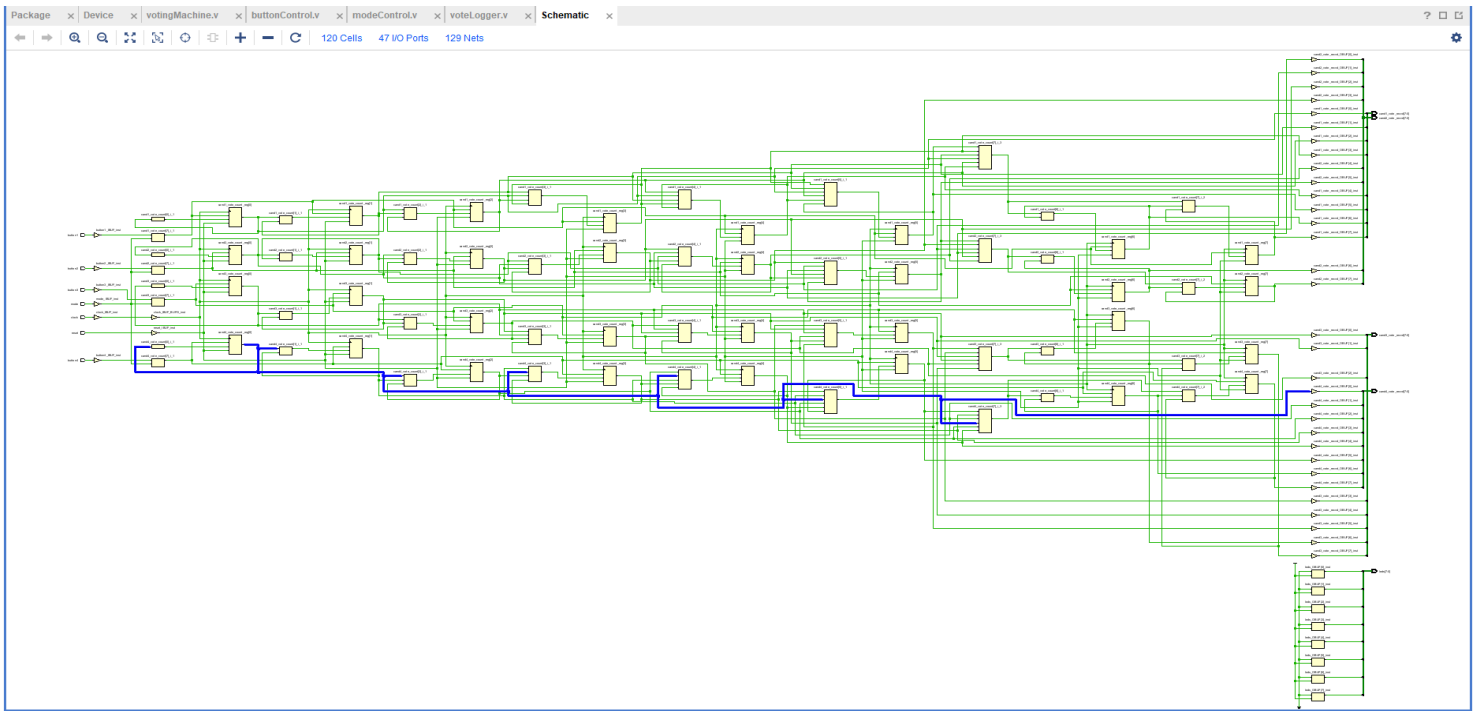
VERILOG CODE

The Verilog Code for those 4 design System is attach in the Zip Folder.



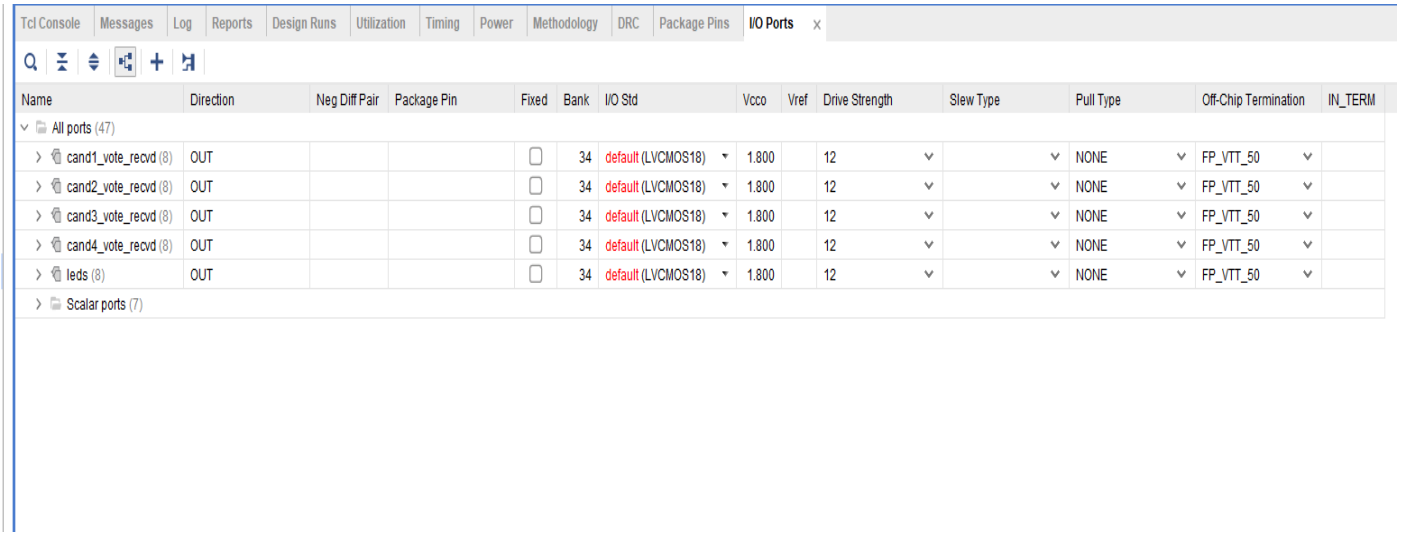
SCHEMATIC AFTER IMPLEMENTATION





Showcasing Various Connecting Lines

IO PORT ASSIGNMENT



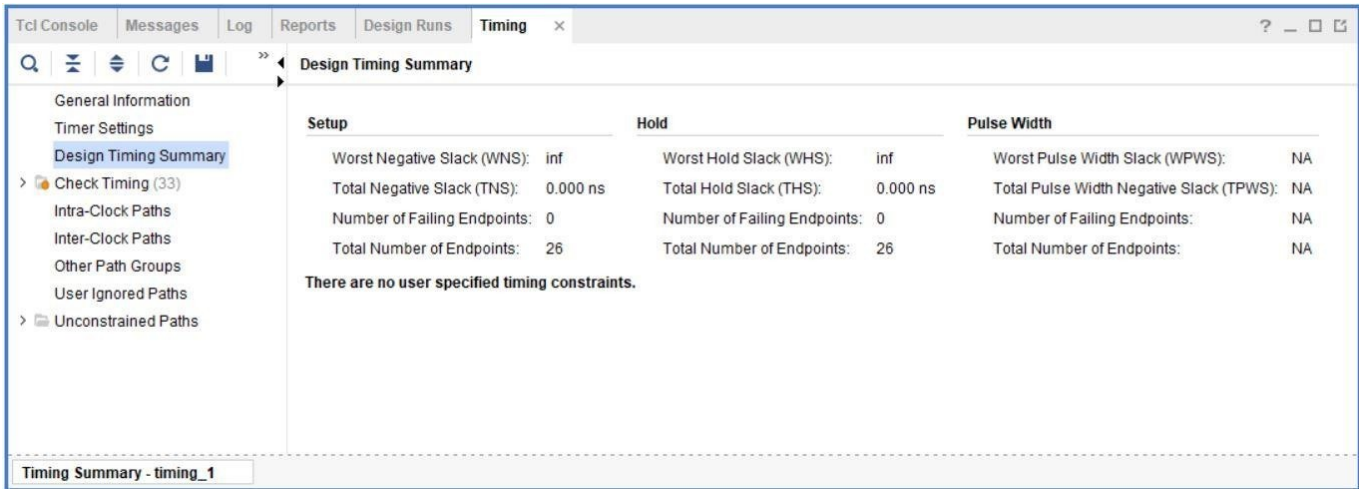
Tcl Console | Messages | Log | Reports | Design Runs | Utilization | Timing | Power | Methodology | DRC | Package Pins | I/O Ports

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
All ports (47)													
cand1_vote_recvd (8)	OUT				34	default (LVCMOS18)	1.800		12		NONE	FP_VTT_50	
cand2_vote_recvd (8)	OUT				34	default (LVCMOS18)	1.800		12		NONE	FP_VTT_50	
cand3_vote_recvd (8)	OUT				34	default (LVCMOS18)	1.800		12		NONE	FP_VTT_50	
cand4_vote_recvd (8)	OUT				34	default (LVCMOS18)	1.800		12		NONE	FP_VTT_50	
leds (8)	OUT				34	default (LVCMOS18)	1.800		12		NONE	FP_VTT_50	
Scalar ports (7)													

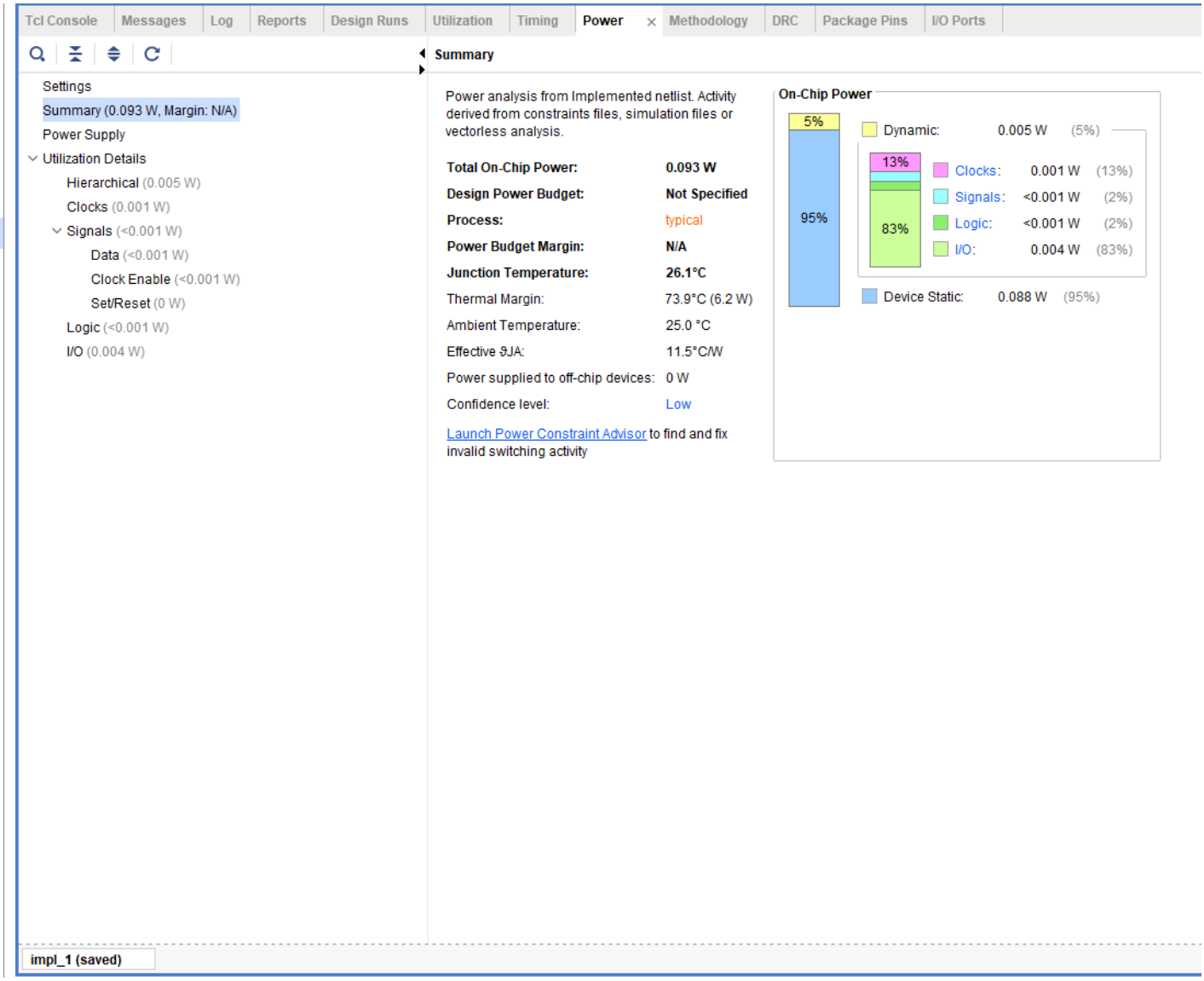
SCHEMATIC AFTER SYNTHESIS

REPORTS AFTER SYNTHESIS

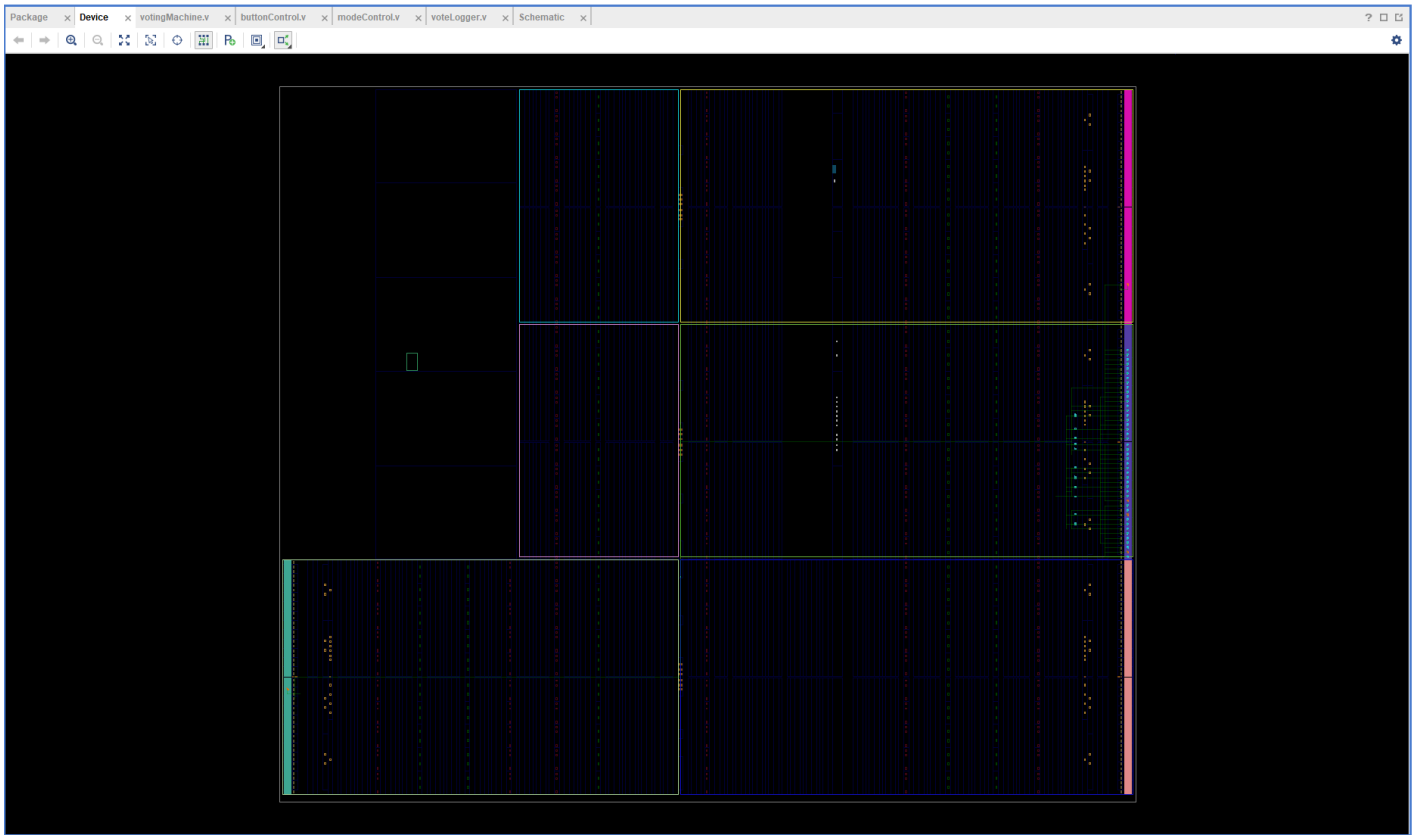
TIMING REPORT

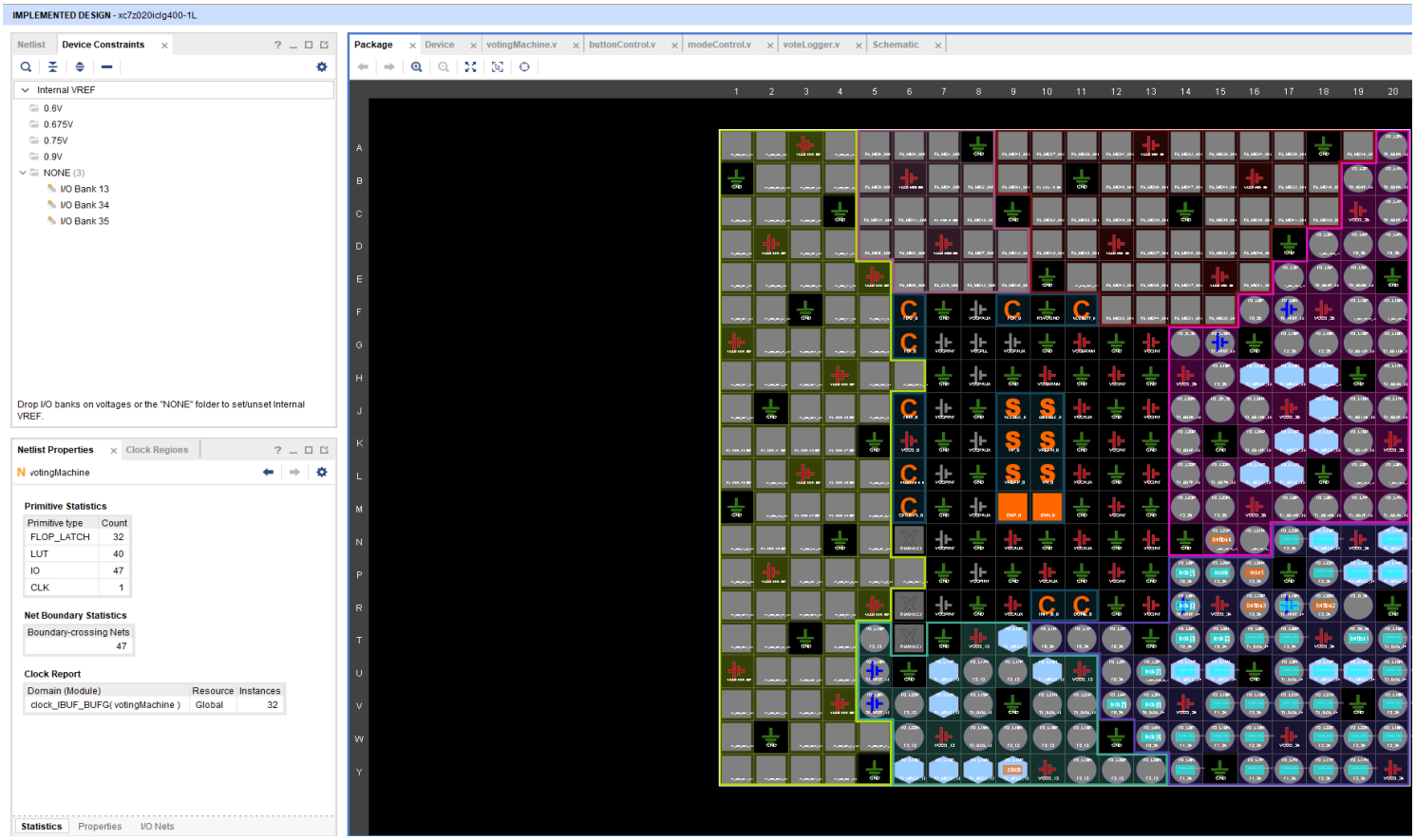


POWER REPORT



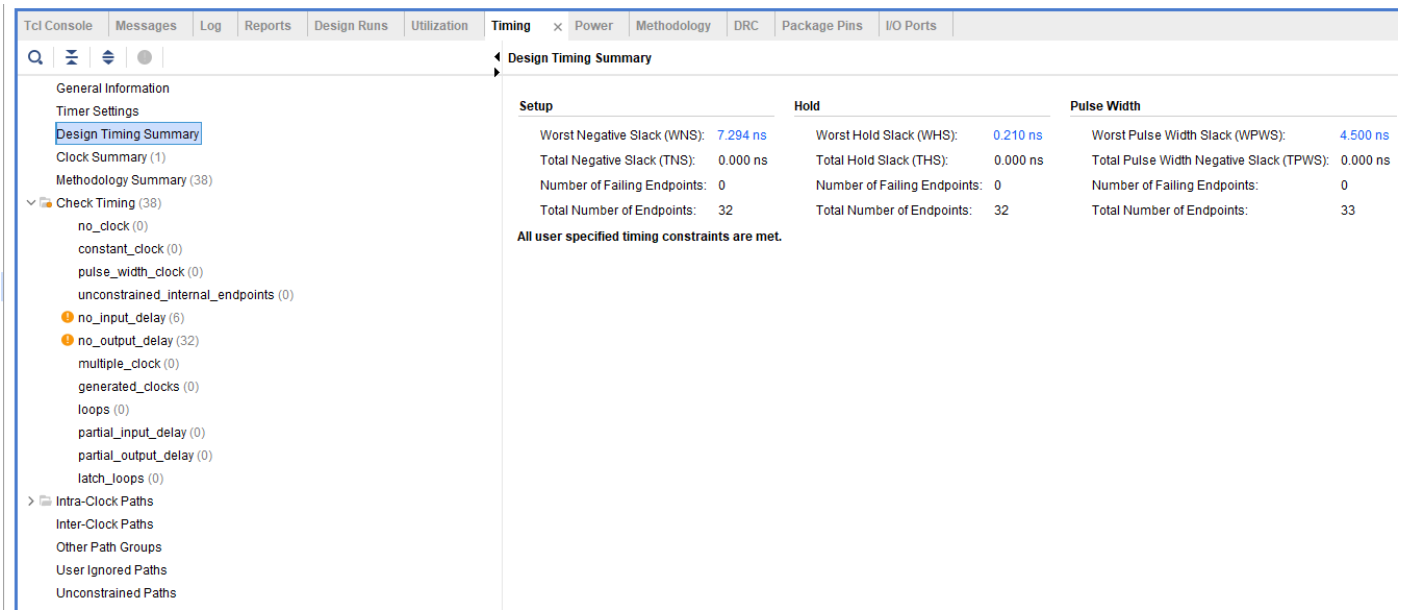
DEVICE LAYOUT AFTER IMPLEMENTATION



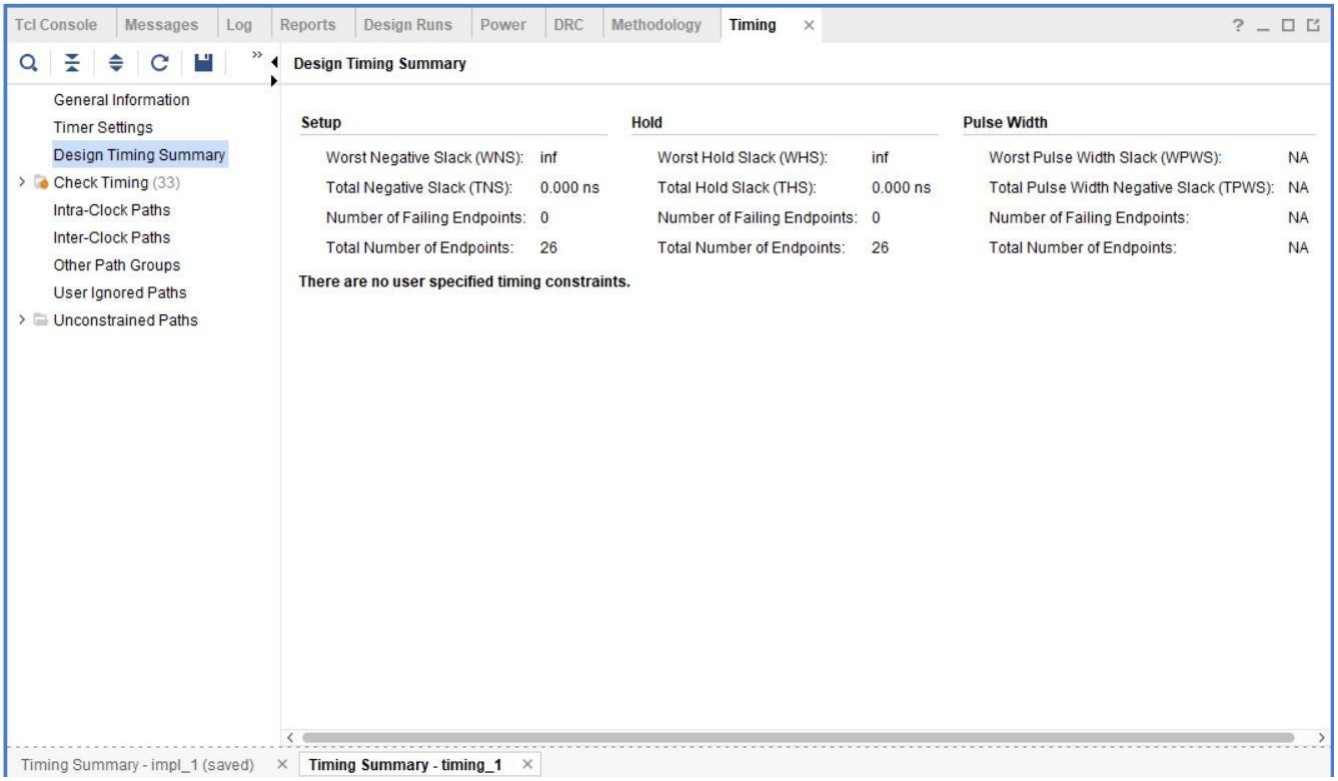


REPORTS AFTER IMPLEMENTATION

TIMING REPORT



OVERALL SYSTEM REPORT

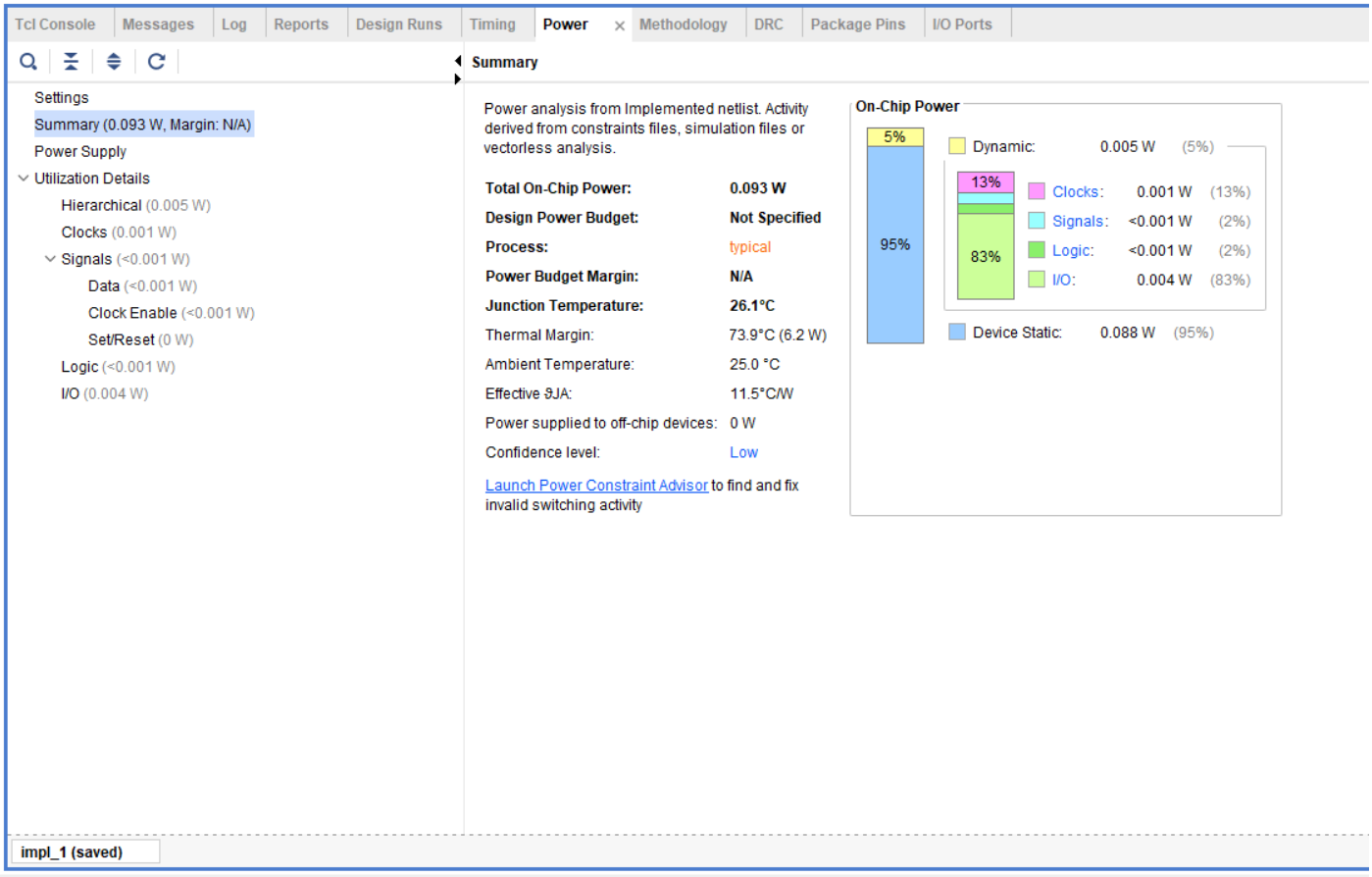


UTILIZATION REPORT

Tcl Console | Messages | Log | Reports | Design Runs | Utilization | Timing | Power | Methodology | DRC | Package Pins | I/O Ports

Report	Type	Options	Modified	Size
Synthesis				
Synth Design (synth_design)				
Utilization - Synth Design	report_utilization		4/28/24, 11:37 AM	7.8 KB
synthesis_report			4/28/24, 11:37 AM	26.7 KB
Implementation				
impl_1				
Design Initialization (init_design)				
Timing Summary	report_timing_summary	max_paths = 10;		
Opt Design (opt_design)				
DRC	report_drc		4/28/24, 11:40 AM	3.9 KB
Timing Summary	report_timing_summary	max_paths = 10;		
Power Opt Design (power_opt_design)				
Timing Summary	report_timing_summary	max_paths = 10;		
Place Design (place_design)				
IO	report_io		4/28/24, 11:40 AM	120.1 KB
Utilization	report_utilization		4/28/24, 11:40 AM	9.7 KB
Control Sets	report_control_sets	verbose = true;	4/28/24, 11:40 AM	4.2 KB
Incremental Reuse	report_incremental_reuse			
Incremental Reuse	report_incremental_reuse			
Timing Summary	report_timing_summary	max_paths = 10;		
Post-Place Power Opt Design (post_place_power_opt_design)				
Timing Summary	report_timing_summary	max_paths = 10;		
Post-Place Phys Opt Design (phys_opt_design)				
Timing Summary	report_timing_summary	max_paths = 10;		
Route Design (route_design)				
DRC	report_drc		4/28/24, 11:41 AM	3.9 KB
Methodology	report_methodology		4/28/24, 11:41 AM	8.7 KB
Power	report_power		4/28/24, 11:41 AM	8.9 KB
Route Status	report_route_status		4/28/24, 11:41 AM	0.6 KB
Timing Summary	report_timing_summary	max_paths = 10;	4/28/24, 11:41 AM	91.6 KB
Incremental Reuse	report_incremental_reuse			
Clock Utilization	report_clock_utilization		4/28/24, 11:41 AM	11.1 KB
Bus Skew	report_bus_skew	warn_on_violation = true;	4/28/24, 11:41 AM	1.0 KB
implementation_log			4/28/24, 11:41 AM	43.3 KB
Post-Route Phys Opt Design (post_route_phys_opt_design)				
Timing Summary	report_timing_summary	max_paths = 10; warn_on_violation = true;		
Bus Skew	report_bus_skew	warn_on_violation = true;		

POWER REPORT



PACKAGE PINS AFTER IMPLEMENTATION

Tcl Console | Messages | Log | Reports | Design Runs | Timing | Power | Methodology | DRC | **Package Pins** | I/O Ports

Name	Available	Prohibit	Ports	I/O Std	Dir	Vcco	Bank	Bank Type	Bank Performance	DDRMC Only	Byte Group	Type	Diff Pair	Clock	Voltage	Config	System Monitor	Gigabit I/O	MCB	PCI	Min Trace Dly (ps)	Max Trace Dly (ps)	Site	Site Type
V19	0											GND												GND
W2	0											GND												GND
W12	0											GND												GND
Y5	0											GND												GND
Y15	0											GND												GND
G13	0											VCCINT												VCCINT
H12	0											VCCINT												VCCINT
J13	0											VCCINT												VCCINT
K12	0											VCCINT												VCCINT
L13	0											VCCINT												VCCINT
M12	0											VCCINT												VCCINT
N13	0											VCCINT												VCCINT
P12	0											VCCINT												VCCINT
R13	0											VCCINT												VCCINT
J11	0											VCCAUX												VCCAUX
L11	0											VCCAUX												VCCAUX
N9	0											VCCAUX												VCCAUX
P10	0											VCCAUX												VCCAUX
R9	0											VCCAUX												VCCAUX
N11	0											VCCAUX												VCCAUX
G11	0											VCCBRAM												VCCBRAM
H10	0											VCCBRAM												VCCBRAM
G8	0											PSS VCCPLL												VCCPLL
G9	0											PSS VCCAUX												VCCPAUX
F8	0											PSS VCCAUX												VCCPAUX
H8	0											PSS VCCAUX												VCCPAUX
K8	0											PSS VCCAUX												VCCPAUX
M8	0											PSS VCCAUX												VCCPAUX
G7	0											PSS VCCINT												VCCPINT
J7	0											PSS VCCINT												VCCPINT
L7	0											PSS VCCINT												VCCPINT
N7	0											PSS VCCINT												VCCPINT
P8	0											PSS VCCINT												VCCPINT
R7	0											PSS VCCINT												VCCPINT

REFERENCES:

1) HW Components Design on FPGA: Practice -Course Notes
2) Nptel lectures on Digital design by prof. Srinivasan
3) http://www.asic-world.com/tidbits/verilog_fsm.html