

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**

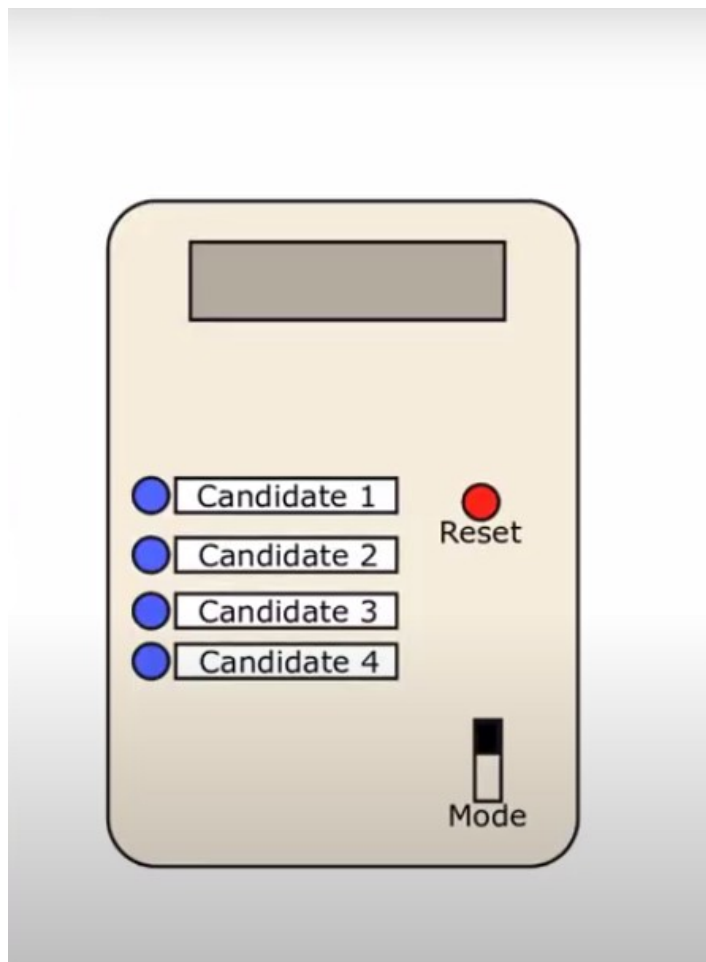
University of Verona (UNVR)

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.

The screenshot displays the Xilinx Vivado Project Manager interface for a project named 'votingMachine'. The 'Sources' pane on the left shows four design sources: **votingMachine** (votingMachine.v), **buttonControl** (buttonControl.v), **modeControl** (modeControl.v), and **voteLogger** (voteLogger.v). The 'Source File Properties' pane for **voteLogger.v** shows it is enabled and located at `C:/Users/FPGA-PROJECT/votingMachine_fpga/votingMachine.srcs/sr`. The 'Project Summary' pane on the right provides details about the project, including the project name, location, family (Zynq-7000), part (xc7z020iclg400-1L), top module name (votingMachine), target language (Verilog), and simulator language (Mixed). The 'Synthesis' pane shows the status as 'Complete' with 16 critical warnings and 34 warnings. The 'Implementation' pane shows the status as 'Not started'. A 'Synthesis Completed' dialog box is overlaid on the interface, indicating that synthesis has successfully completed. The dialog box offers options for the next steps: 'Run Implementation' (selected), 'Open Synthesized Design', 'View Reports', and 'Don't show this dialog again'. The 'Tcl Console' at the bottom shows the progress of the synthesis process, including messages about handling custom attributes, renaming generated nets, and writing the synthesis report. The console also displays a report of cell usage, showing the count of various cells used in the design.

Project Manager - votingMachine

Sources

- Design Sources (4)
 - votingMachine** (votingMachine.v)
 - buttonControl (buttonControl.v)
 - modeControl (modeControl.v)
 - voteLogger (voteLogger.v)
- Constraints (1)
 - constrs_1 (1)

Source File Properties

voteLogger.v

- Enabled
- Location: C:/Users/FPGA-PROJECT/votingMachine_fpga/votingMachine.srcs/sr
- Type: Verilog
- Library: xil_defaultlib

Project Summary

Overview | Dashboard

Settings | Edit

Project name: votingMachine
Project location: C:/Users/FPGA-PROJECT/votingMachine_fpga
Product family: Zynq-7000
Project part: xc7z020iclg400-1L
Top module name: votingMachine
Target language: Verilog
Simulator language: Mixed

Synthesis

Status: Complete
Messages: 16 critical warnings, 34 warnings
Part: xc7z020iclg400-1L
Strategy: Flow_AreaOptimized_high
Report Strategy: Vivado Synthesis Default Reports
Incremental synthesis: Automatically selected checkpoint

Implementation

Status: Not started
Messages: No errors or warnings
Part: xc7z020iclg400-1L
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default Reports
Incremental implementation: None

Synthesis Completed

Information: Synthesis successfully completed.

Next

- ☒ Run Implementation
- ☐ Open Synthesized Design
- ☐ View Reports
- ☐ Don't show this dialog again

OK Cancel

Tcl Console

Finished Handling Custom Attributes : Time (s): cpu = 00:00:40 ; elapsed = 00:00:44 . Memory (MB): peak = 1530.289 ; gain = ...

Start Renaming Generated Nets

Finished Renaming Generated Nets : Time (s): cpu = 00:00:40 ; elapsed = 00:00:44 . Memory (MB): peak = 1530.289 ; gain = 637

Start Writing Synthesis Report

Report BlackBoxes:

BlackBox name	Instances
IBUF	1
I2C1	4
I2C2	12
I2C3	8
I2C4	4
I2C5	4
I2C6	8
I2C7	32
I2C8	7
I2C9	32
I2C10	8

Report Cell Usage:

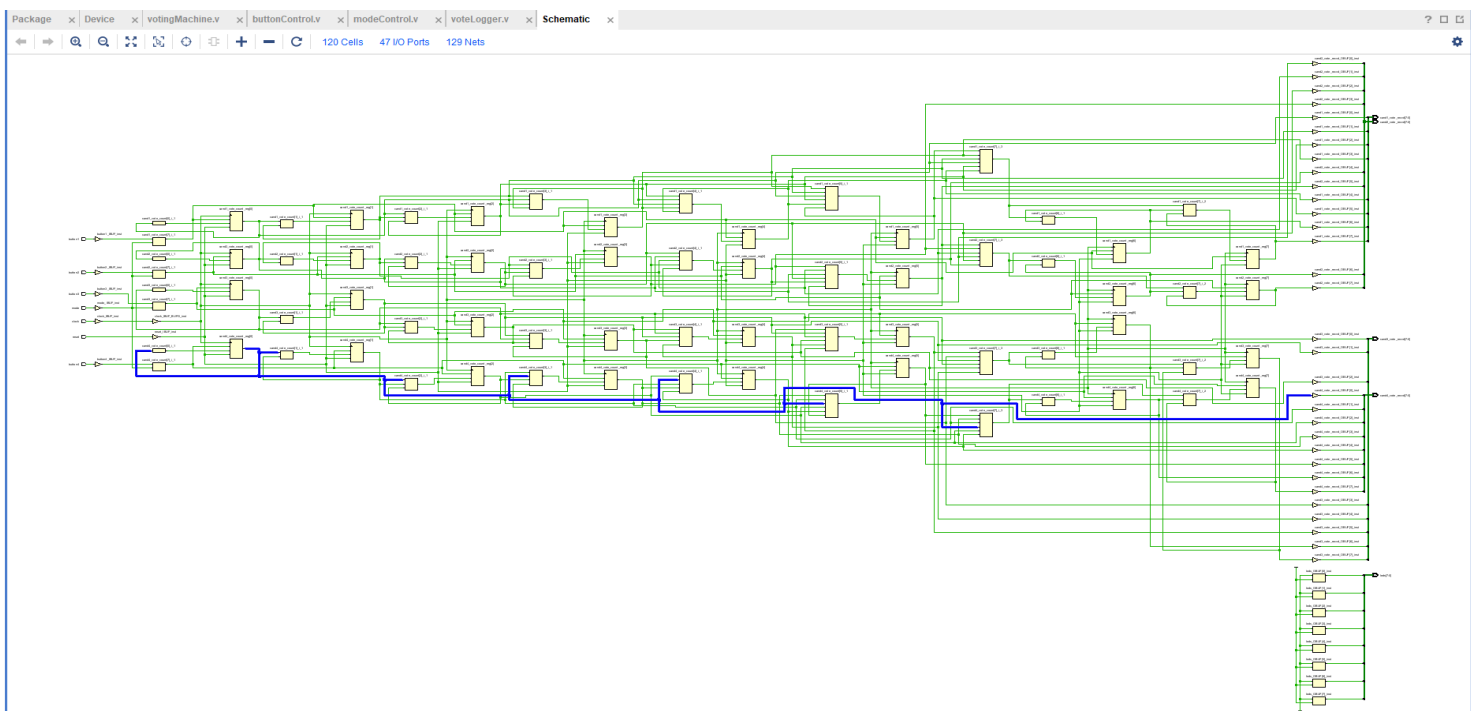
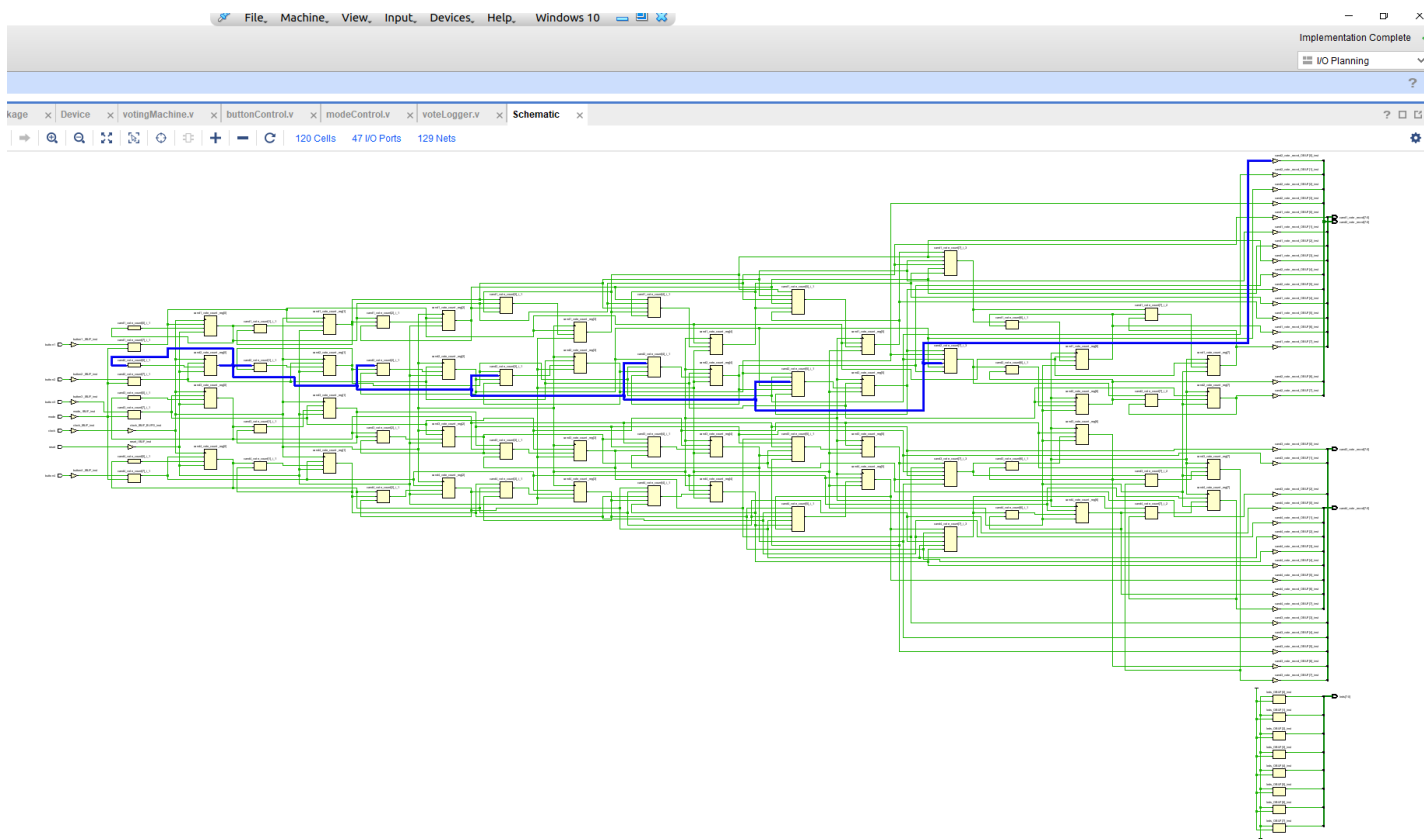
Cell	Count
IBUF	1
I2C1	4
I2C2	12
I2C3	8
I2C4	4
I2C5	4
I2C6	8
I2C7	32
I2C8	7
I2C9	32
I2C10	8

VERILOG CODE

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

[illegible]

SCHEMATIC AFTER IMPLEMENTATION



Showcasing Various Connecting Lines

IO PORT ASSIGNMENT

Tcl ConsoleMessagesLogReportsDesign RunsUtilizationTimingPowerMethodologyDRCPackage PinsI/O Ports x

Q

↔

⬇

⬆

+

⬇

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_rcvd (8)	OUT			<input type="checkbox"/>	34	default (LVCMOS18) ▼	1.800		12 ▼	▼	NONE ▼	FP_VTT_50 ▼	▼
> cand2_vote_rcvd (8)	OUT			<input type="checkbox"/>	34	default (LVCMOS18) ▼	1.800		12 ▼	▼	NONE ▼	FP_VTT_50 ▼	▼
> cand3_vote_rcvd (8)	OUT			<input type="checkbox"/>	34	default (LVCMOS18) ▼	1.800		12 ▼	▼	NONE ▼	FP_VTT_50 ▼	▼
> cand4_vote_rcvd (8)	OUT			<input type="checkbox"/>	34	default (LVCMOS18) ▼	1.800		12 ▼	▼	NONE ▼	FP_VTT_50 ▼	▼
> leds (8)	OUT			<input type="checkbox"/>	34	default (LVCMOS18) ▼	1.800		12 ▼	▼	NONE ▼	FP_VTT_50 ▼	▼
> Scalar ports (7)													

SCHEMATIC AFTER SYNTHESIS

REPORTS AFTER SYNTHESIS

TIMING REPORT

Tcl Console	Messages	Log	Reports	Design Runs	Timing x	?	—	□	↗
Design Timing Summary									
<div>General Information</div> <div>Timer Settings</div> <div>Design Timing Summary</div> <div>> Check Timing (33)</div> <div>Intra-Clock Paths</div> <div>Inter-Clock Paths</div> <div>Other Path Groups</div> <div>User Ignored Paths</div> <div>> Unconstrained Paths</div>									
Setup			Hold			Pulse Width			
Worst Negative Slack (WNS): inf			Worst Hold Slack (WHS): inf			Worst Pulse Width Slack (WPWS): NA			
Total Negative Slack (TNS): 0.000 ns			Total Hold Slack (THS): 0.000 ns			Total Pulse Width Negative Slack (TPWS): NA			
Number of Failing Endpoints: 0			Number of Failing Endpoints: 0			Number of Failing Endpoints: NA			
Total Number of Endpoints: 26			Total Number of Endpoints: 26			Total Number of Endpoints: NA			
There are no user specified timing constraints.									
Timing Summary - timing_1									

POWER REPORT

Tcl ConsoleMessagesLogReportsDesign RunsUtilizationTimingPower xMethodologyDRCPackage PinsI/O Ports

Q⏏⚙️🔄

Summary

Settings

Summary (0.093 W, Margin: N/A)

Power Supply

Utilization Details

Hierarchical (0.005 W)

Clocks (0.001 W)

Signals (<0.001 W)

Data (<0.001 W)

Clock Enable (<0.001 W)

Set/Reset (0 W)

Logic (<0.001 W)

I/O (0.004 W)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:0.093 W

Design Power Budget:Not Specified

Process:typical

Power Budget Margin:N/A

Junction Temperature:26.1°C

Thermal Margin:73.9°C (6.2 W)

Ambient Temperature:25.0 °C

Effective θJA:11.5°C/W

Power supplied to off-chip devices: 0 W

Confidence level:Low

Launch Power Constraint Advisor

to find and fix invalid switching activity

On-Chip Power

5%95%

Dynamic:0.005 W (5%)

13%83%

Clocks:0.001 W (13%)

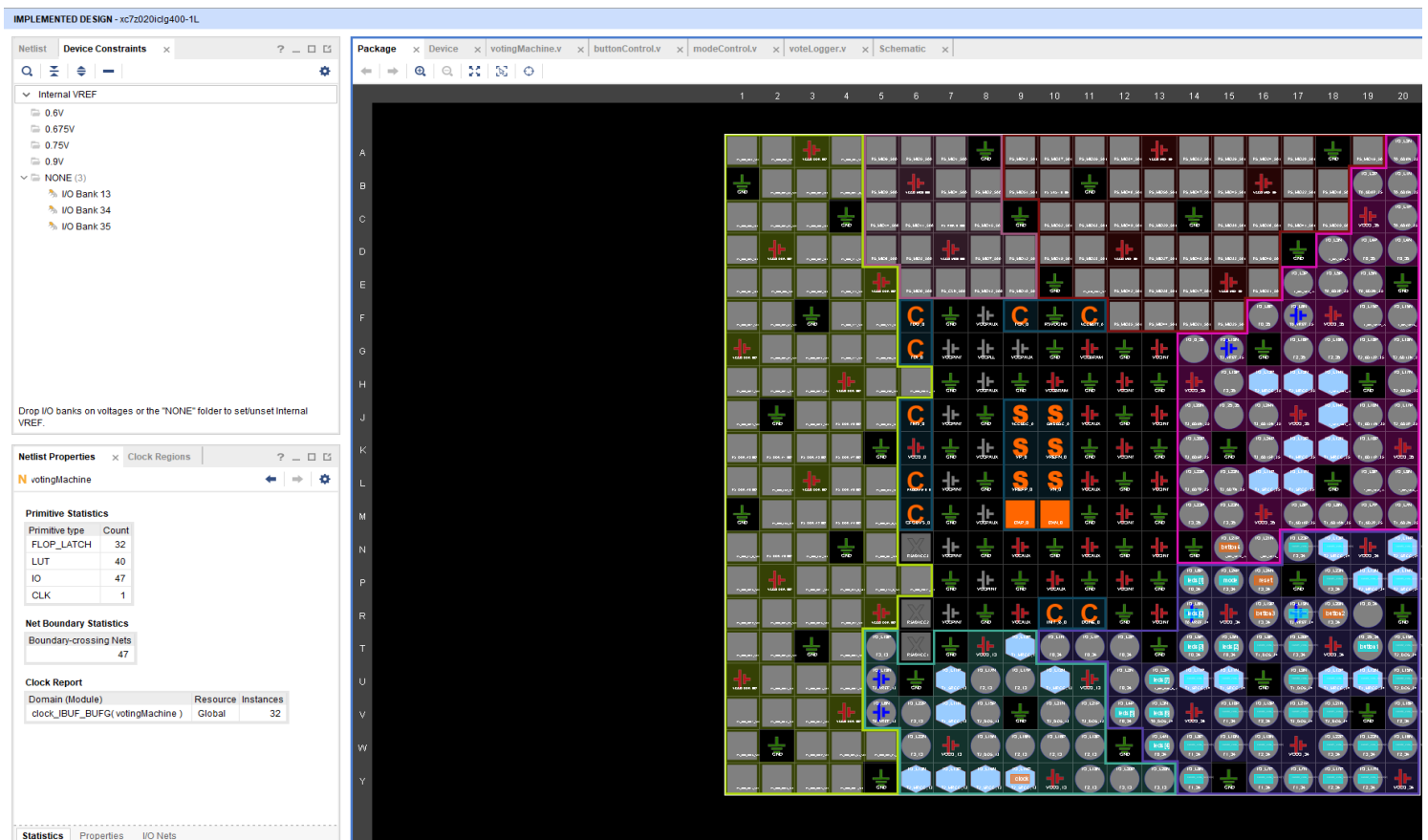
Signals:<0.001 W (2%)

Logic:<0.001 W (2%)

I/O:0.004 W (83%)

Device Static:0.088 W (95%)

impl_1 (saved)



TIMING REPORT

Tcl Console | Messages | Log | Reports | Design Runs | Utilization

Timing x Power Methodology DRC Package Pins I/O Ports

🔍 ⌵ ⚙️ ●

General Information

Timer Settings

Design Timing Summary

Clock Summary (1)

Methodology Summary (38)

▼ Check Timing (38)

- no_clock (0)
- constant_clock (0)
- pulse_width_clock (0)
- unconstrained_internal_endpoints (0)
- ❗ no_input_delay (6)
- ❗ no_output_delay (32)
- multiple_clock (0)
- generated_clocks (0)
- loops (0)
- partial_input_delay (0)
- partial_output_delay (0)
- latch_loops (0)

> Intra-Clock Paths

- Inter-Clock Paths
- Other Path Groups
- User Ignored Paths
- Unconstrained Paths

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 7.294 ns	Worst Hold Slack (WHS): 0.210 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 32	Total Number of Endpoints: 32	Total Number of Endpoints: 33

All user specified timing constraints are met.

OVERALL SYSTEM REPORT

Tcl Console

Messages

Log

Reports

Design Runs

Power

DRC

Methodology

Timing x

?

—

□

✕

Q

≡

≡

↺

📁

»

Design Timing Summary

General Information

Timer Settings

Design Timing Summary

Check Timing (33)

Intra-Clock Paths

Inter-Clock Paths

Other Path Groups

User Ignored Paths

Unconstrained Paths

Setup

Hold

Pulse Width

Worst Negative Slack (WNS): inf

Worst Hold Slack (WHS): inf

Worst Pulse Width Slack (WPWS): NA

Total Negative Slack (TNS): 0.000 ns

Total Hold Slack (THS): 0.000 ns

Total Pulse Width Negative Slack (TPWS): NA

Number of Failing Endpoints: 0

Number of Failing Endpoints: 0

Number of Failing Endpoints: NA

Total Number of Endpoints: 26

Total Number of Endpoints: 26

Total Number of Endpoints: NA

There are no user specified timing constraints.

Timing Summary - impl_1 (saved) x

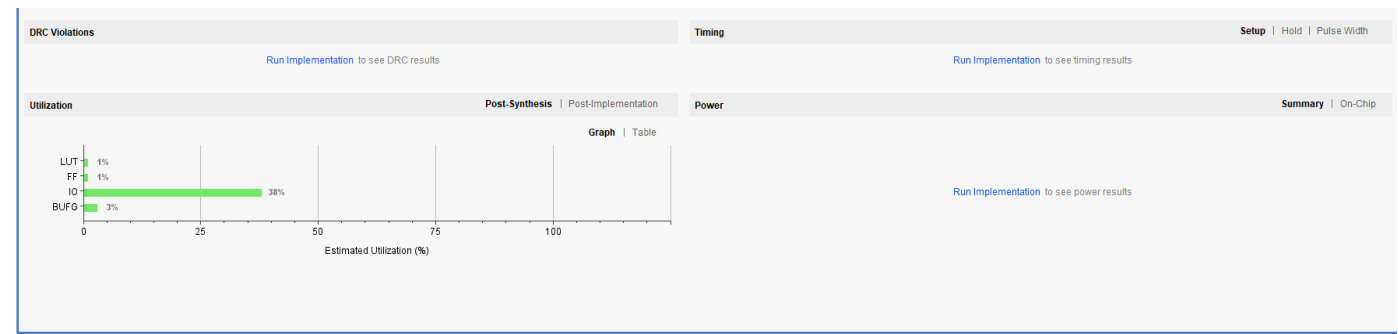
Timing Summary - timing_1 x

UTILIZATION REPORT

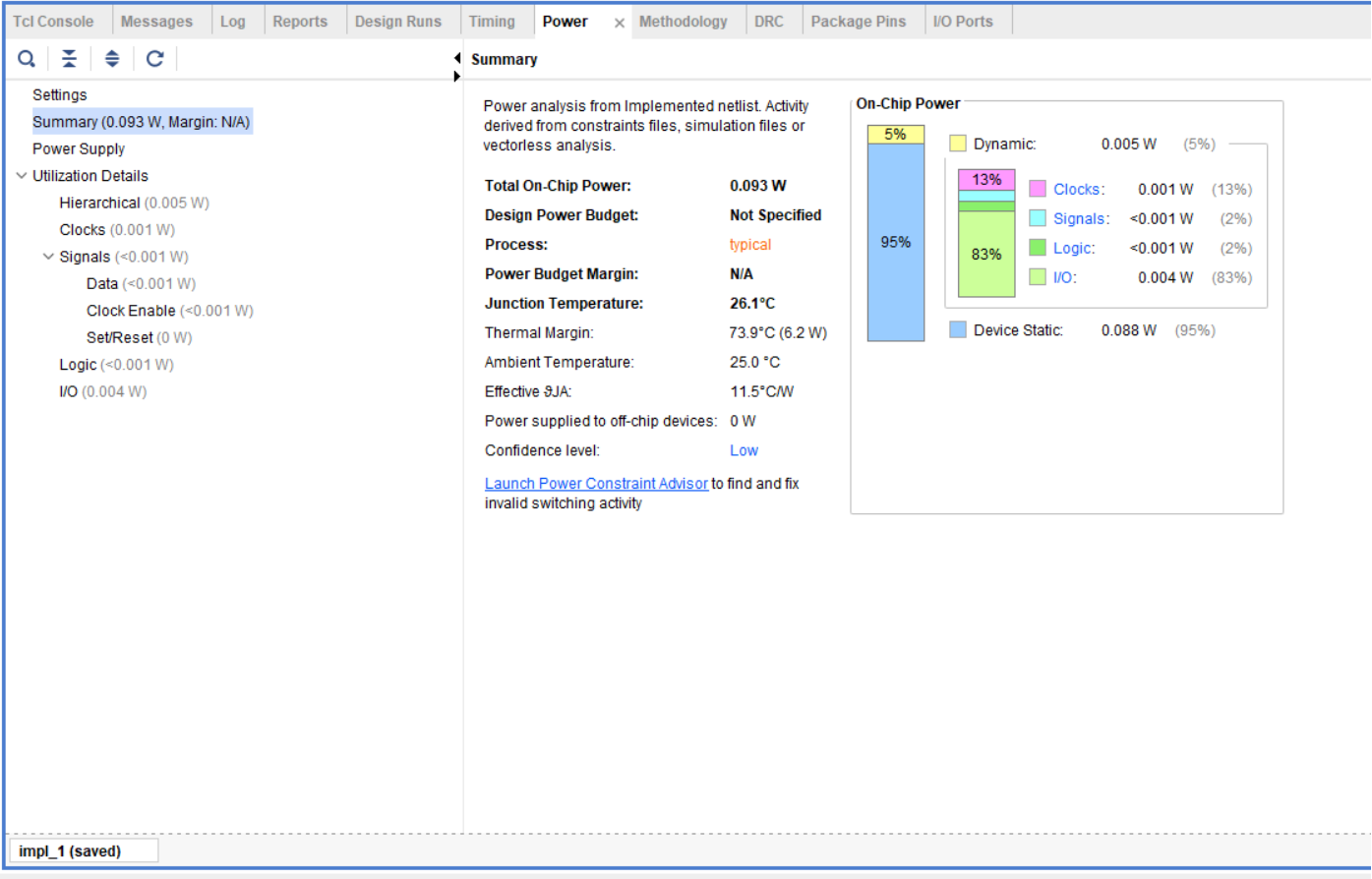
Tcl ConsoleMessagesLogReportsDesign RunsUtilizationTimingPowerMethodologyDRCPackage PinsI/O Ports

Q⏮⏪⏩⏭+−✎▶

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

[illegible]

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