MACHINE LEARNING ARCHITECTURES

Course Project

Hardware Implementation of Matrix Convolution with Configurable Padding and Stride Length

Submitted by,

AILA TEJ RITHVIK - 244102401

EEE - VLSI & NANOELECTRONICS

Contents

1 Introduction 2

2 Matrix Convolution Overview 2

3 Architecture 2

4 Design Considerations 3

5 RTL Schematic 4
5.1 Fixed Point 16-bit 4
5.2 Floating Point 32-bit 5

6 Behavioral Simulation 6
6.1 Fixed Point 16-bit 6
6.2 Floating Point 32-bit 7

7 Synthesis Results (Kintex-7 FPGA) 8
7.1 Fixed point 16-bit 8
7.1.1 Utilization 8
7.1.2 Power Consumption 9
7.1.3 Timing Report 10
7.2 Floating point 32-bit 10
7.2.1 Utilization 10
7.2.2 Power Consumption 11
7.2.3 Timing Report 11

8 Post Synthesis Functional Simulation 12
8.1 Fixed Point 16-bit 12
8.2 Floating Point 32-bit 13

9 Post Synthesis Timing Simulation 14
9.1 Fixed Point 16-bit 14
9.2 Floating Point 32-bit 15

10 Conclusion 15

1 Introduction

Convolution is a fundamental operation in signal and image processing, widely used in machine learning, particularly in convolutional neural networks (CNNs). Hardware acceleration of convolution is essential for real-time image recognition, edge processing, and embedded AI systems. This project presents a hardware implementation of a configurable 2D matrix convolution that supports variable image size, filter size, padding, and stride length. Inputs are provided sequentially for both the image matrix and the filter matrix, making the design scalable and resource-efficient.

The architecture is developed in Verilog HDL and synthesized for FPGA deployment. The design enables trade-offs between computation time, area, and flexibility.

2 Matrix Convolution Overview

2D Convolution involves sliding a filter (kernel) over an image and computing the dot product at each position. The output dimensions depend on:

- **Image Size** $(H \times W)$
- **Filter Size** $(F \times F)$
- **Padding** (P)
- **Stride Length** (S)

The output feature map dimensions are given by:

$$O_H = \frac{H - F + 2P}{S} + 1, \quad O_W = \frac{W - F + 2P}{S} + 1 \tag{1}$$

3 Architecture

The hardware design is controlled by a Finite State Machine (FSM) with the following states:

- **IDLE:** The system waits for a start signal and initializes all control signals.
- **READ_INPUTS:** Sequentially reads image pixels, filter coefficients, and configuration parameters (image size, filter size, padding, stride).
- **OPERATE:** Performs the core computations such as convolution and intermediate processing.
- **DISPLAY_OUTPUTS:** Transfers the computed feature map data to the output buffer for sequential access.
- **DONE:** Indicates the completion of processing and waits for reset or the next start signal.

4 Design Considerations

Key considerations in the design include:

- **Sequential Input Feeding:** Eliminates the need for large parallel input buses, making the design area-efficient.
- **Configurable Parameters:** Image size, filter size, padding, and stride are programmable.
- **Fixed-Point Arithmetic:** Ensures hardware-friendly implementation with reduced resource utilization.

5 RTL Schematic

5.1 Fixed Point 16-bit

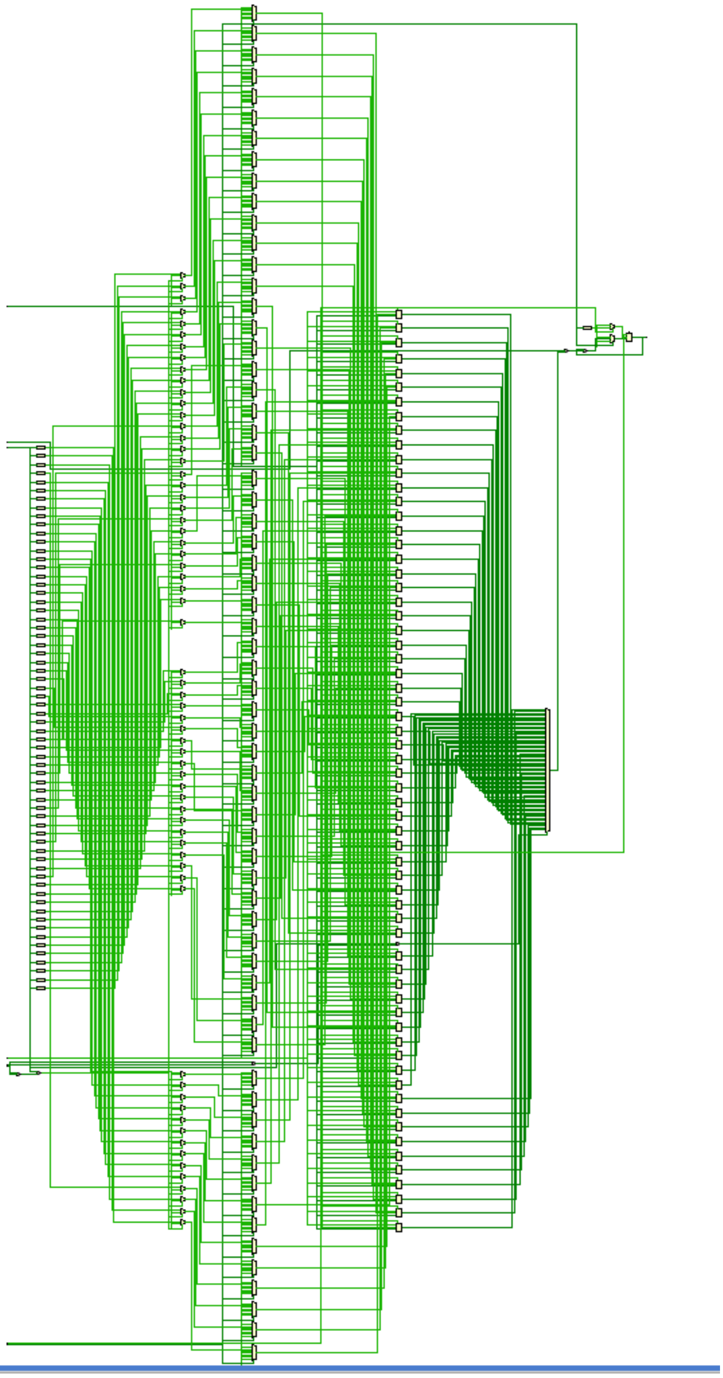


Figure 1: RTL Schematic of Matrix Convolution (Fixed point 16-bit)

5.2 Floating Point 32-bit

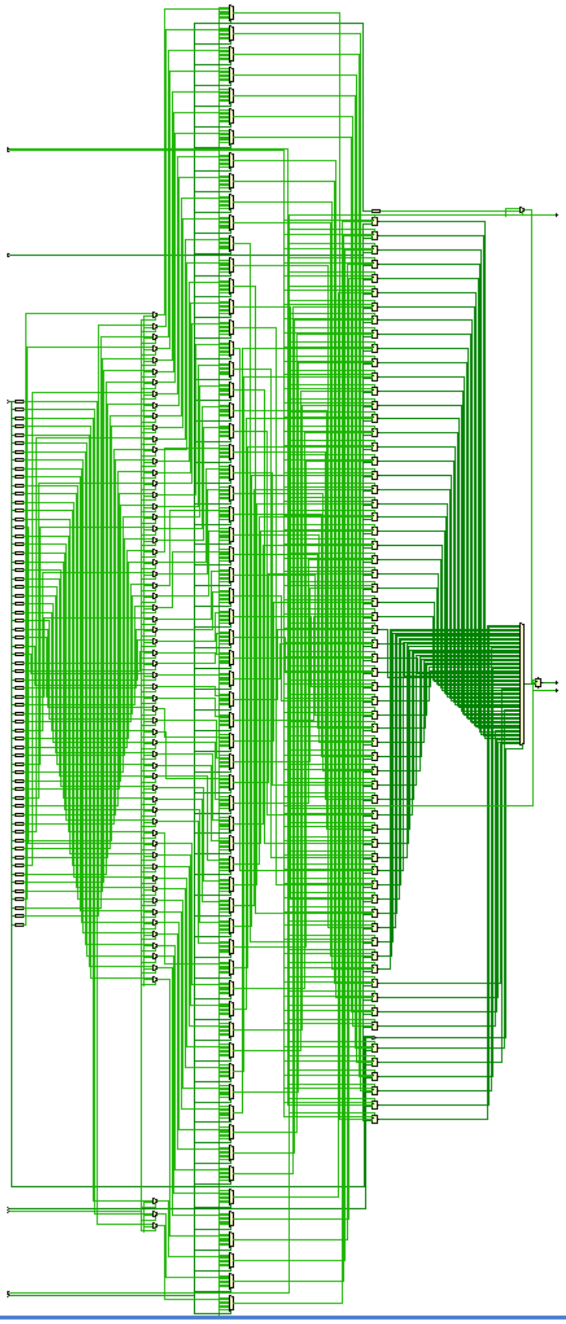


Figure 2: RTL Schematic of Matrix Convolution (Floating point 32-bit)

6 Behavioral Simulation

6.1 Fixed Point 16-bit

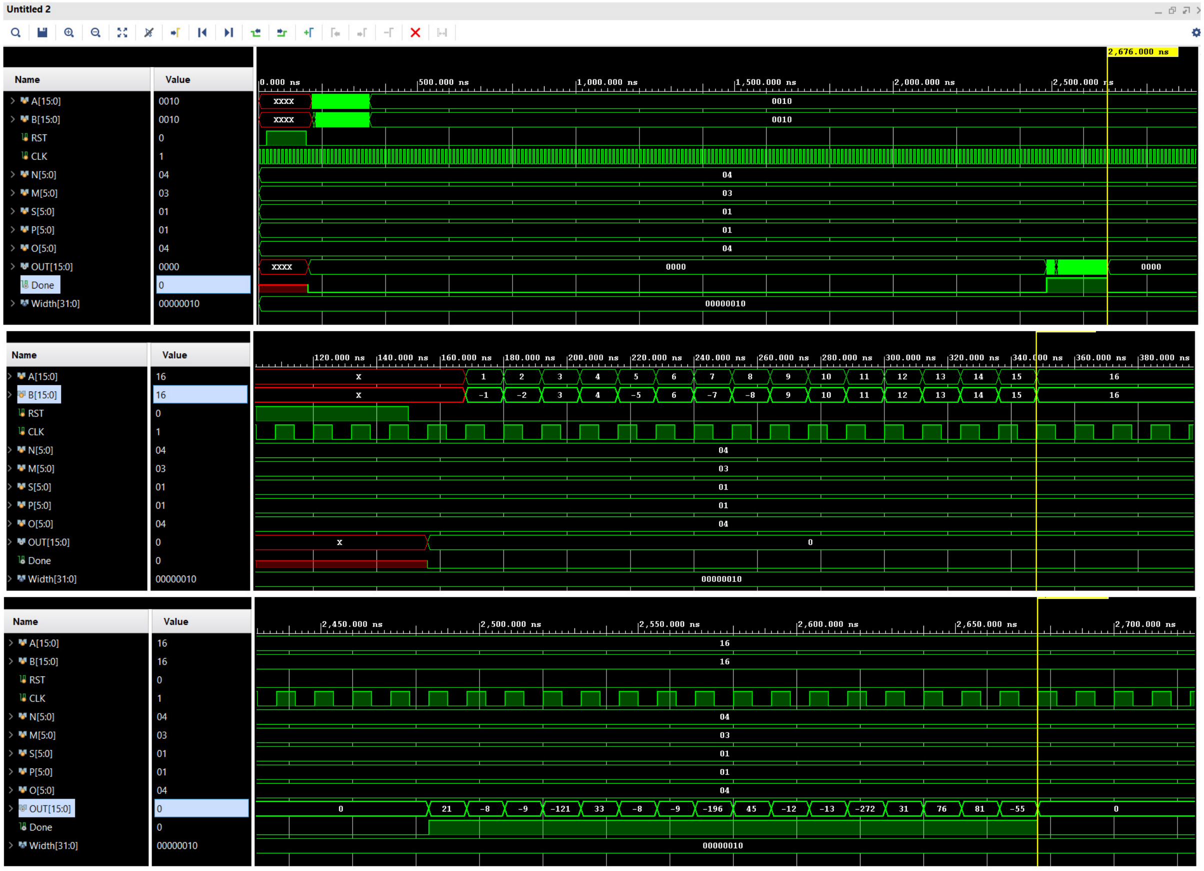


Figure 3: Sequential Input Feeding of Image and Filter on top and reading outputs in the below half

6.2 Floating Point 32-bit

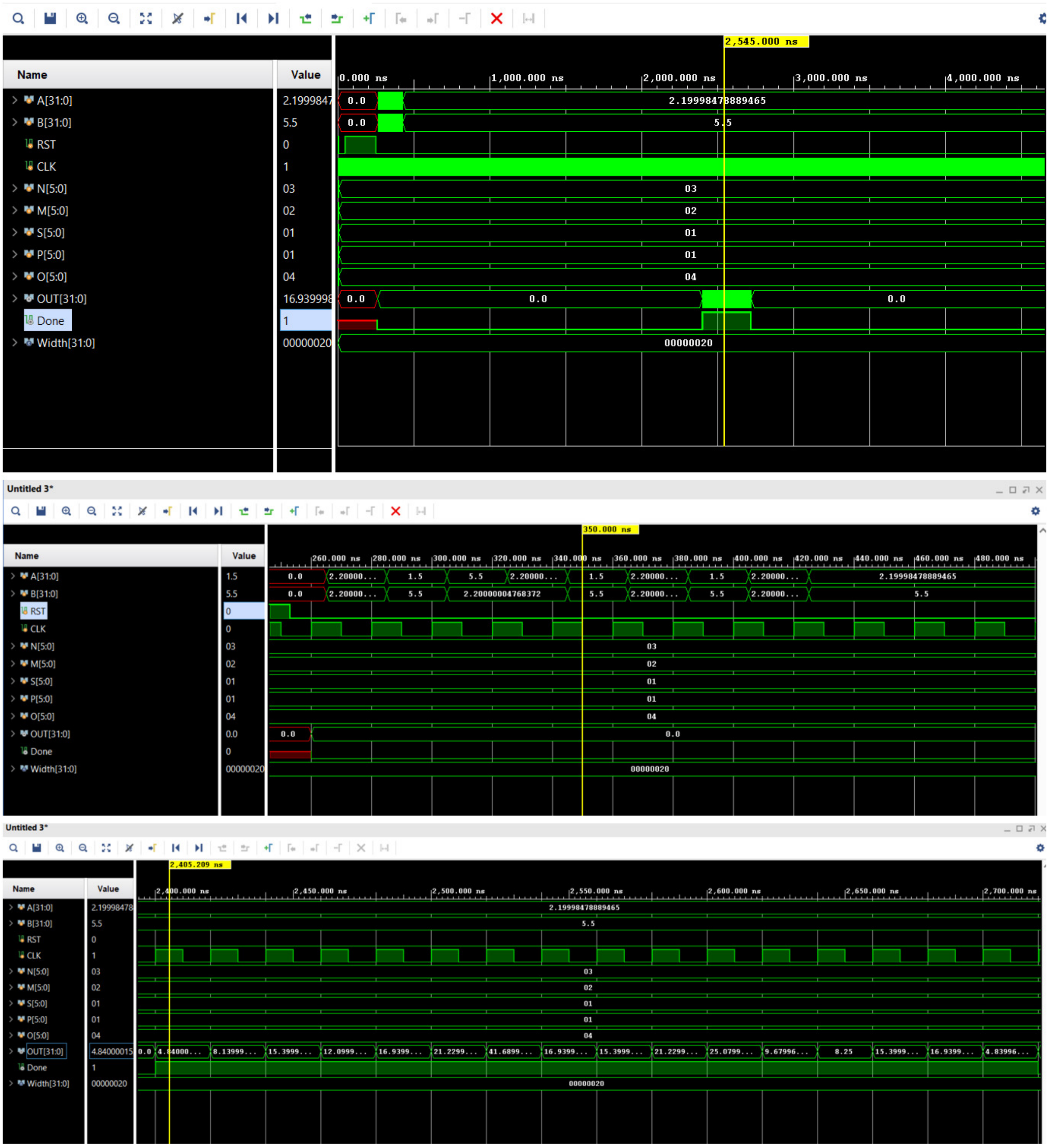
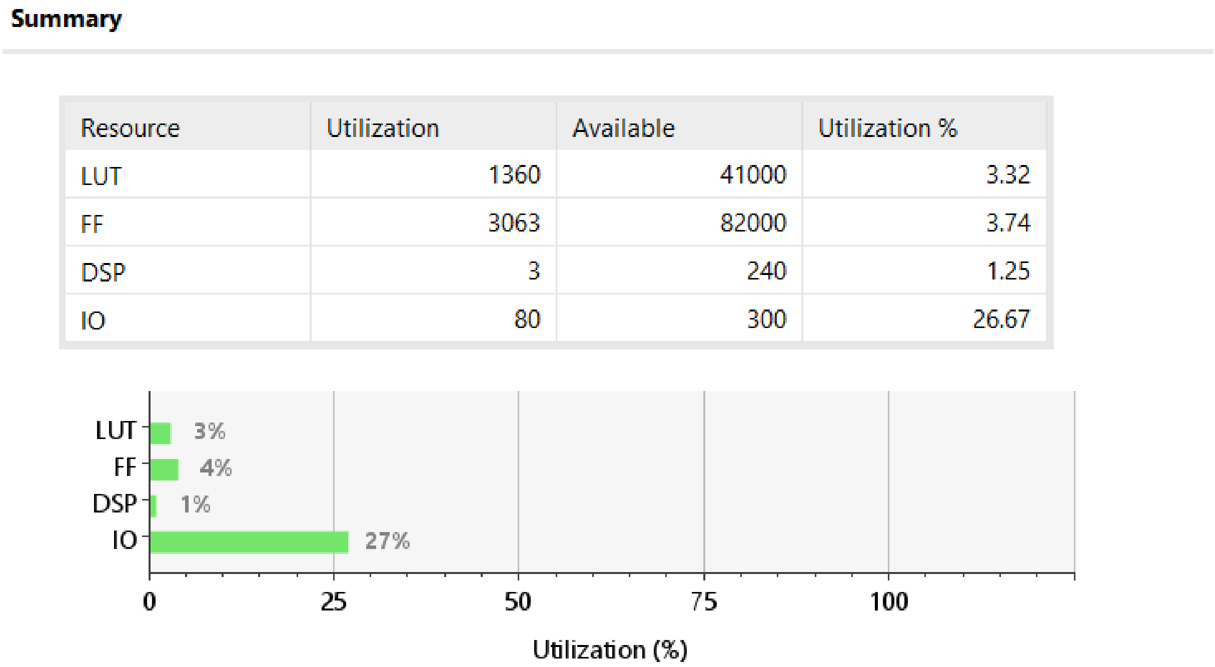


Figure 4: Sequential Input Feeding of Image and Filter on top and reading outputs in the below half

7 Synthesis Results (Kintex-7 FPGA)

7.1 Fixed point 16-bit

7.1.1 Utilization

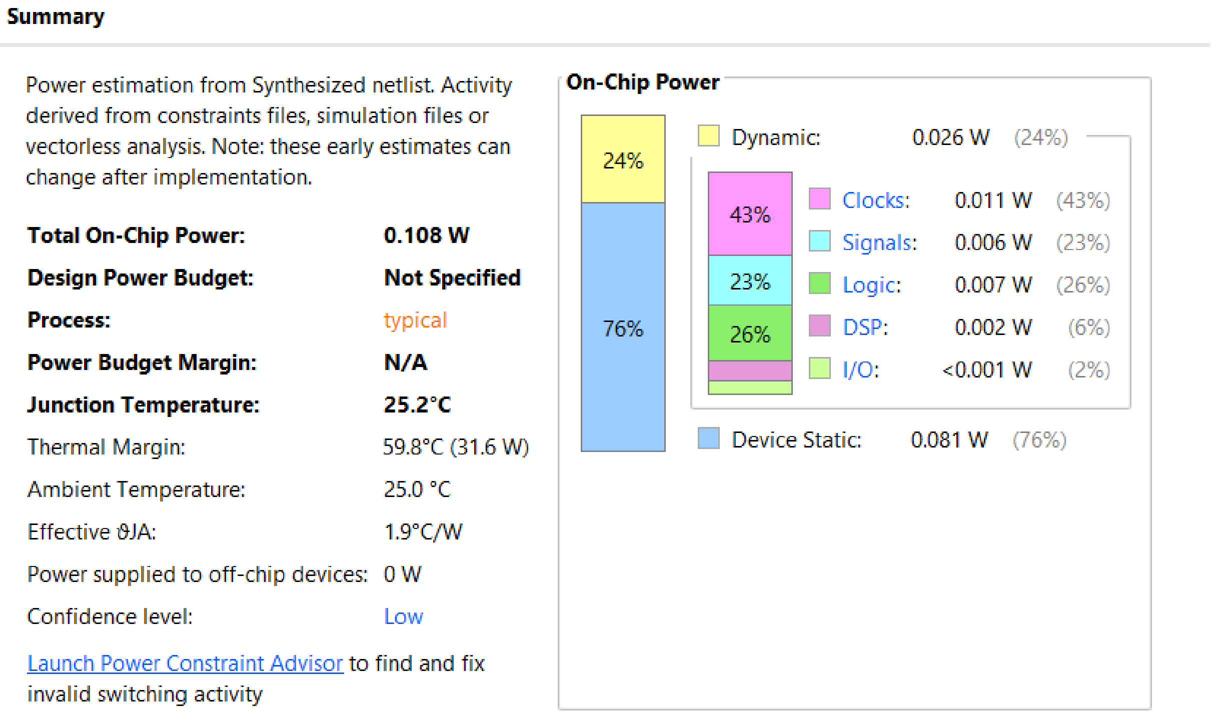


Summary

Resource	Utilization	Available	Utilization %
LUT	1360	41000	3.32
FF	3063	82000	3.74
DSP	3	240	1.25
IO	80	300	26.67

Figure 5: Resource Utilization Report

7.1.2 Power Consumption



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.108 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**25.2°C**
Thermal Margin:	59.8°C (31.6 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

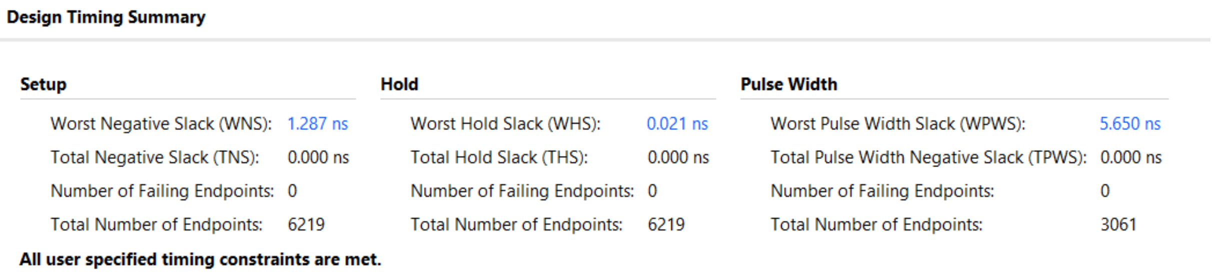
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

🟨	Dynamic:	0.026 W	(24%)
🟪	Clocks:	0.011 W	(43%)
🟦	Signals:	0.006 W	(23%)
🟩	Logic:	0.007 W	(26%)
🟪	DSP:	0.002 W	(6%)
🟩	I/O:	<0.001 W	(2%)
🟦	Device Static:	0.081 W	(76%)

Figure 6: Power Consumption Report

7.1.3 Timing Report



Design Timing Summary

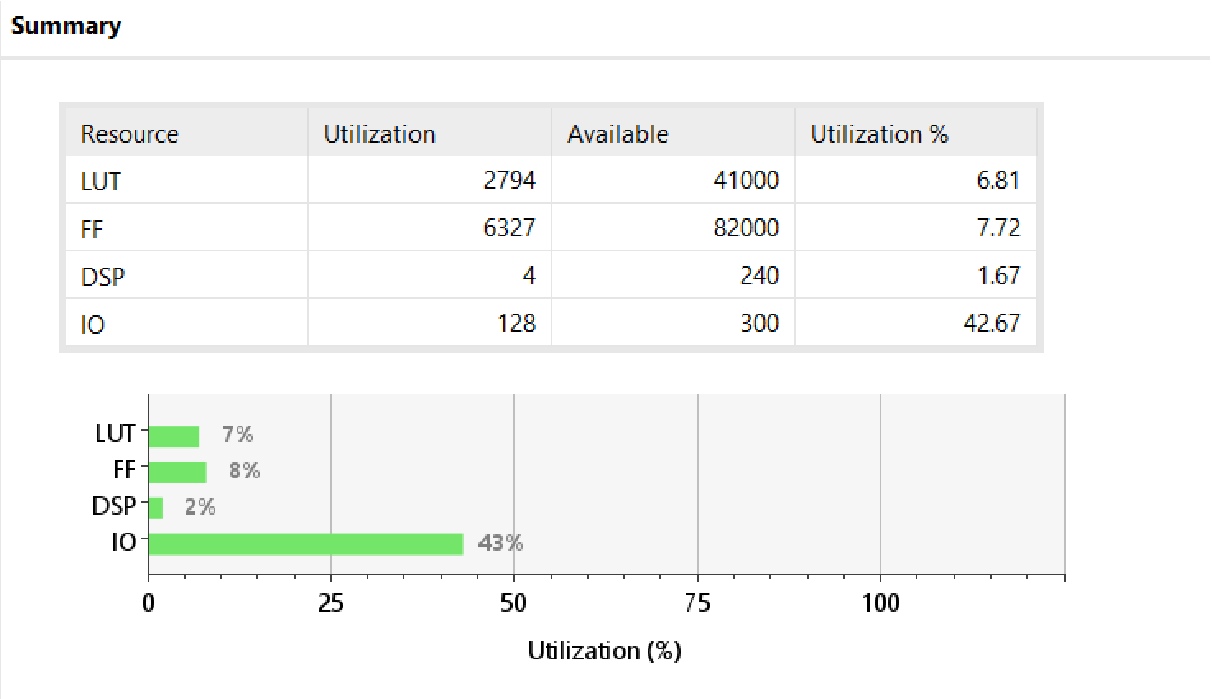
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	1.287 ns	Worst Hold Slack (WHS):	0.021 ns	Worst Pulse Width Slack (WPWS):	5.650 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	6219	Total Number of Endpoints:	6219	Total Number of Endpoints:	3061

All user specified timing constraints are met.

Figure 7: Timing Summary

7.2 Floating point 32-bit

7.2.1 Utilization

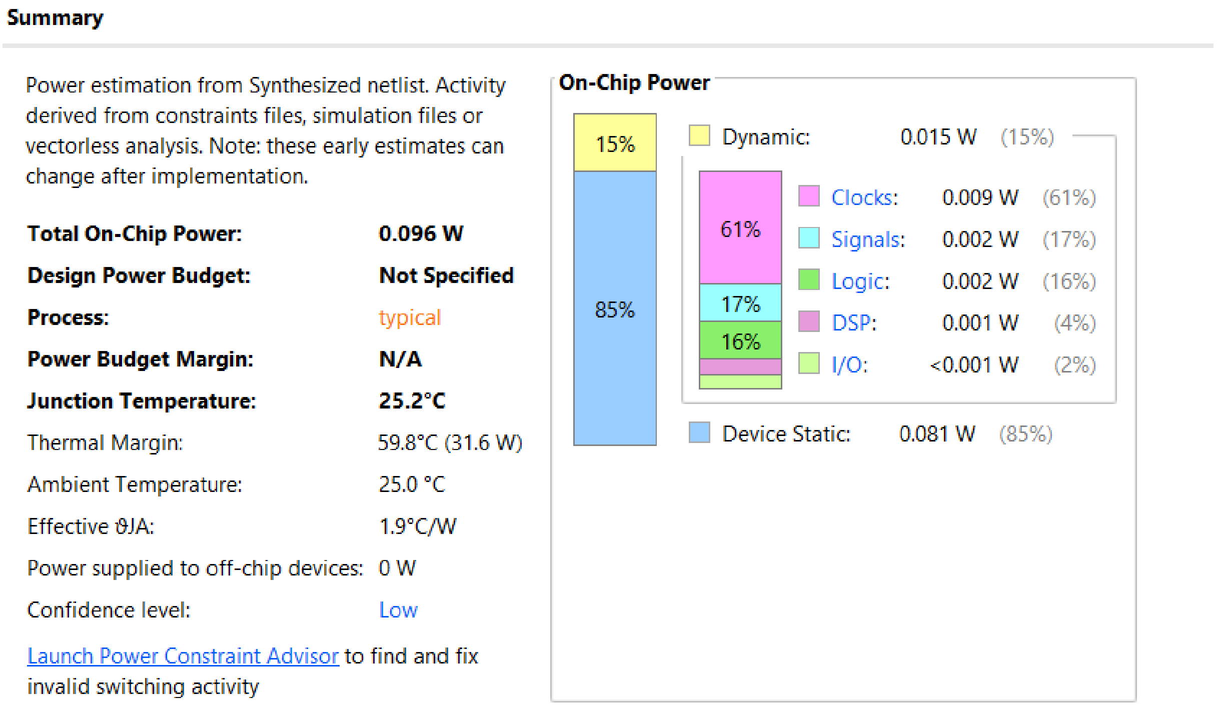


Summary

Resource	Utilization	Available	Utilization %
LUT	2794	41000	6.81
FF	6327	82000	7.72
DSP	4	240	1.67
IO	128	300	42.67

Figure 8: Resource Utilization Report

7.2.2 Power Consumption



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.096 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**25.2°C**
Thermal Margin:	59.8°C (31.6 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

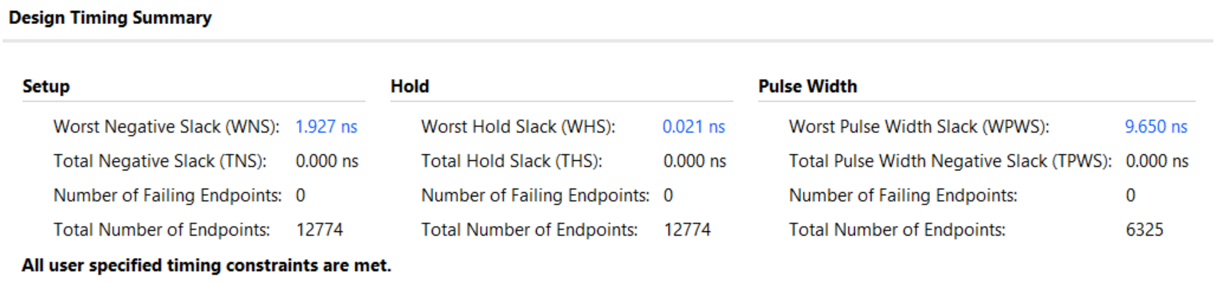
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

🟨	Dynamic:	0.015 W	(15%)
🟪	Clocks:	0.009 W	(61%)
🟦	Signals:	0.002 W	(17%)
🟩	Logic:	0.002 W	(16%)
🟪	DSP:	0.001 W	(4%)
🟩	I/O:	<0.001 W	(2%)
🟦	Device Static:	0.081 W	(85%)

Figure 9: Power Consumption Report

7.2.3 Timing Report



Design Timing Summary

Setup		**Hold**		**Pulse Width**	
Worst Negative Slack (WNS):	1.927 ns	Worst Hold Slack (WHS):	0.021 ns	Worst Pulse Width Slack (WPWS):	9.650 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:	12774	Total Number of Endpoints:	12774	Total Number of Endpoints:	6325

All user specified timing constraints are met.

Figure 10: Timing Summary

8 Post Synthesis Functional Simulation

8.1 Fixed Point 16-bit

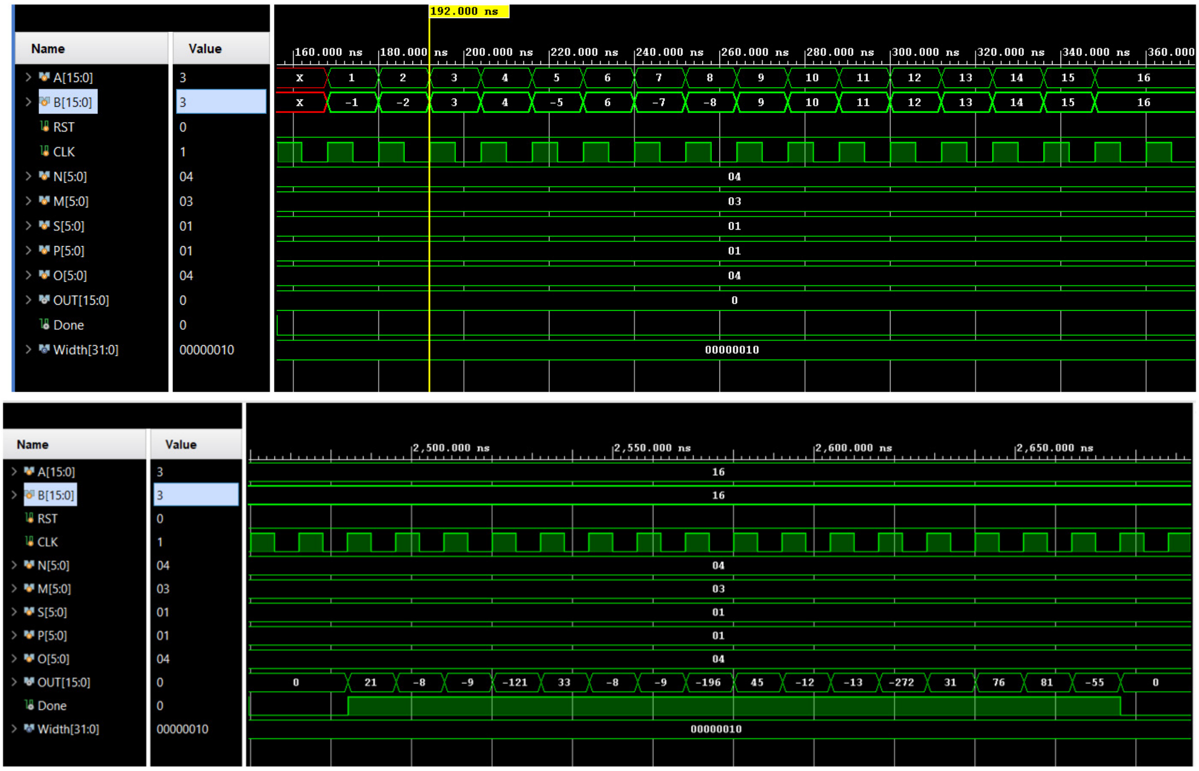


Figure 11: Sequential Input Feeding of Image and Filter on top and reading outputs in the below half

8.2 Floating Point 32-bit

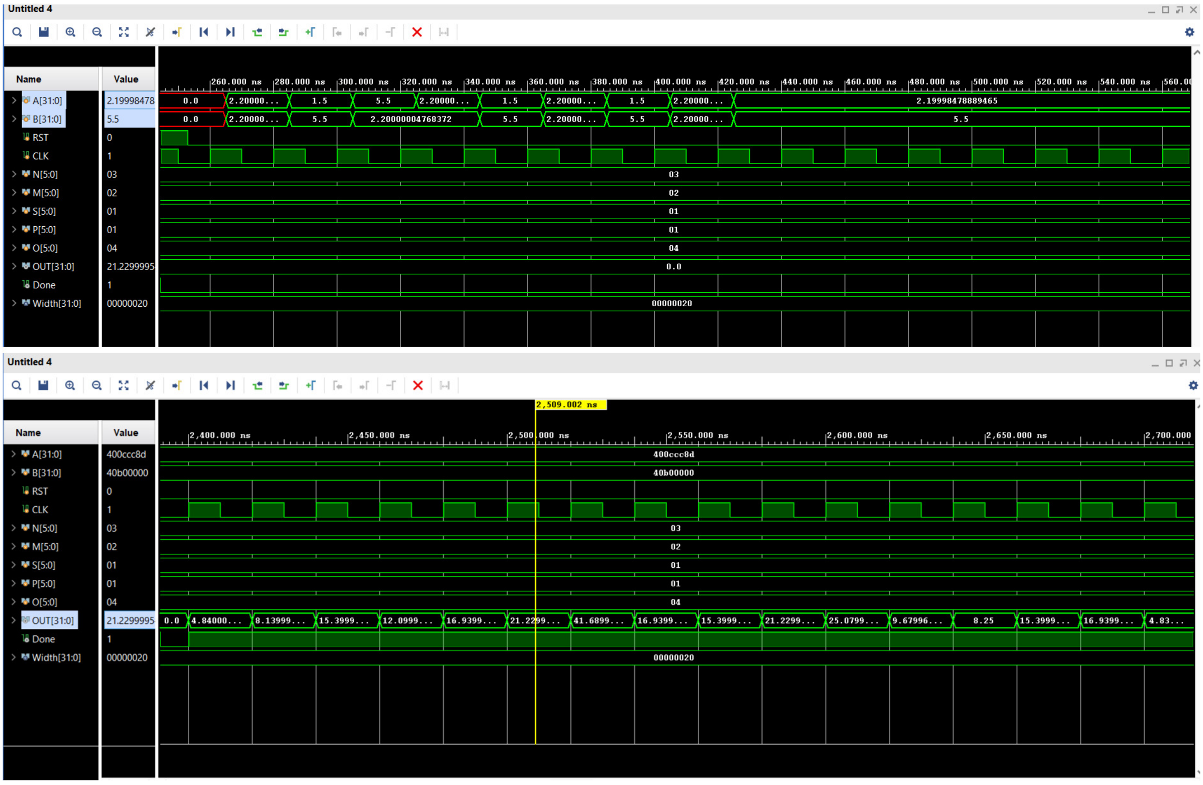


Figure 12: Sequential Input Feeding of Image and Filter on top and reading outputs in the below half

9 Post Synthesis Timing Simulation

9.1 Fixed Point 16-bit

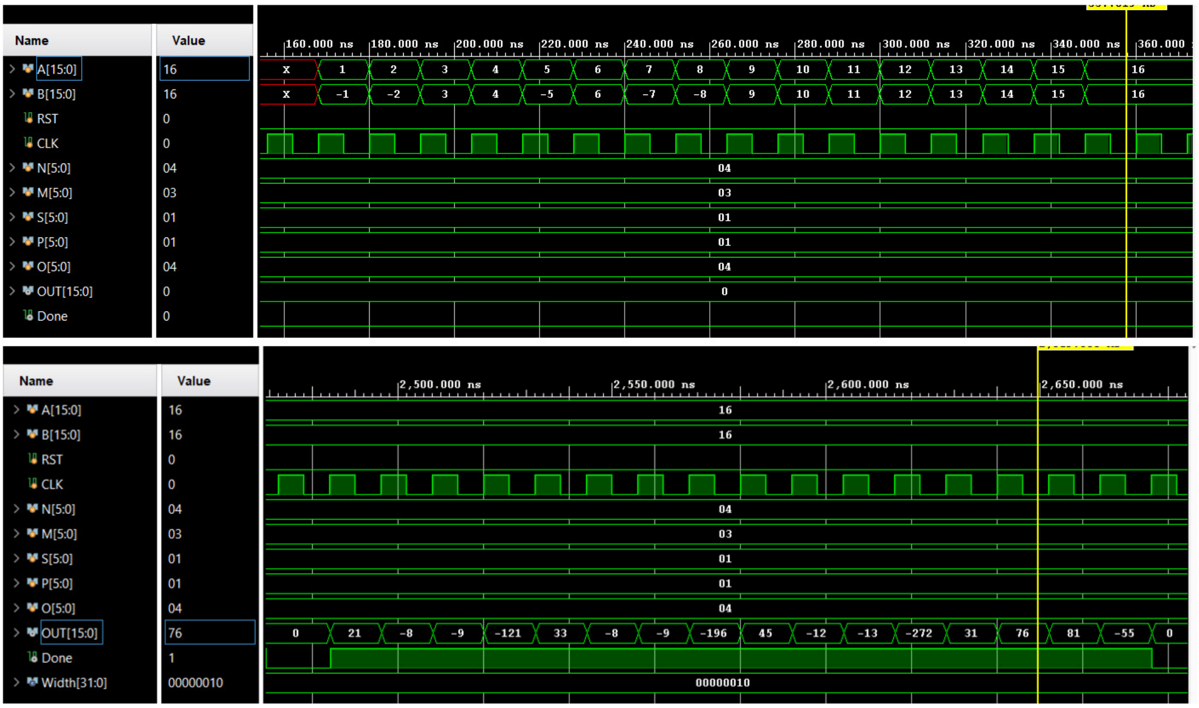


Figure 13: Sequential Input Feeding of Image and Filter on top and reading outputs in the below half

9.2 Floating Point 32-bit

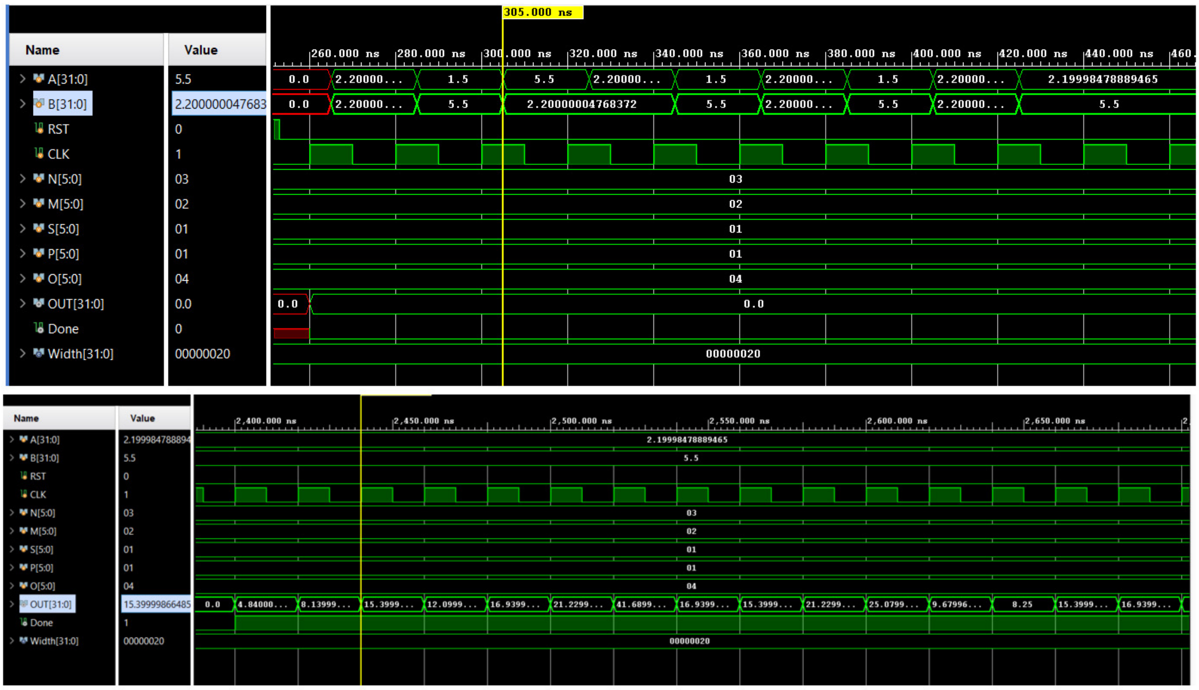


Figure 14: SSequential Input Feeding of Image and Filter on top and reading outputs in the below half

10 Conclusion

A hardware implementation of 2D matrix convolution with configurable image size, filter size, padding, and stride length was successfully designed and verified. The sequential input feeding approach reduces hardware complexity while retaining flexibility. Synthesis results on a Kintex-7 FPGA show the design achieves efficient resource utilization and meets timing requirements, making it suitable for real-time embedded vision and machine learning applications.