
# EE 566 System on Chip Design Spring 2024 Lab2

**Yiğit Bektaş GÜRSOY**

**S042073**

**Class Lecturer: İlker Hamzaoğlu**

# 1. Verilog Source Files

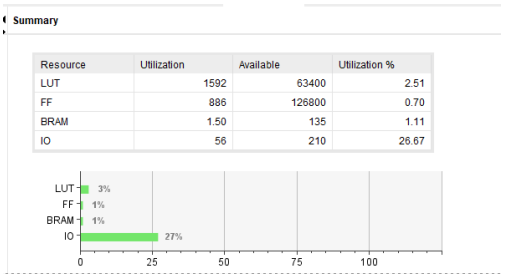
- **SearchWindowMemory.v:**
  Implements the memory that stores the search window data used in the motion estimation process.
- **ReferenceBlockMemory.v:**
  Stores the reference block data that is compared against the search window data during motion estimation.
- **ControlUnit.v:**
  Manages the control signals for coordinating the operations of different components in the motion estimation hardware.
- **Datapath.v:**
  Defines the data path for processing the input data, performing the necessary computations for motion estimation.
- **InstantMinComp.v:**
  Computes the minimum value instantaneously, used for determining the best match in motion estimation.
- **MinTrackerComp.v:**
  Tracks the minimum values over multiple comparisons to find the optimal motion vector.
- **MotionEstimationTop.v:**
  Top-level module that integrates all submodules and coordinates the overall motion estimation process.
- **ProcessingElements.v:**
  Contains the array of processing elements that perform the core computations for motion estimation, such as calculating the Sum of Absolute Differences (SAD).

# 2. Verilog Testbench Files

- **randblock_gen.py:**
  Generates random blocks for use in motion estimation testing.
- **min_SAD.py:**
  Computes the minimum Sum of Absolute Differences (SAD) in software for comparison with hardware results.
- **ME_tb.v:**
  Testbench for the Motion Estimation hardware. It initializes signals, loads data from text files into memories, and compares hardware and software results.
- **SearchWindowMemory_hw.txt:**
  Contains binary data for initializing the search window memory in the testbench.
- **ReferenceBlock_hw.txt:**
  Contains binary data for initializing the reference block memory in the testbench.
- **SearchWindowMemory_sw.txt:**
  Contains decimal data for initializing the search window memory in the software.
- **ReferenceBlock_sw.txt:**
  Contains decimal data for initializing the reference block memory in the software.
- **min_SAD.txt:**
  Contains the expected minimum SAD value calculated by the software for comparison with the hardware result.

# 3. Reports

**FPGA MODEL: Nexys-A7-100T**

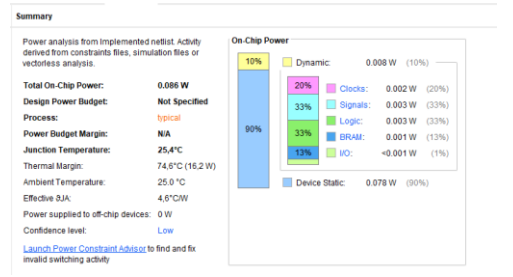


Summary

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 1592 | 63400 | 2.51 |
| FF | 886 | 126800 | 0.70 |
| BRAM | 1.50 | 135 | 1.11 |
| IO | 56 | 210 | 26.67 |



Utilization Report



Summary

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

| | |
|---|---|
| Total On-Chip Power: | 0.086 W |
| Design Power Budget: | Not Specified |
| Process: | typical |
| Power Budget Margin: | N/A |
| Junction Temperature: | 25,4°C |
| Thermal Margin: | 74,6°C (16,2 W) |
| Ambient Temperature: | 25,0 °C |
| Effective θJA: | 4,6°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



Power Report

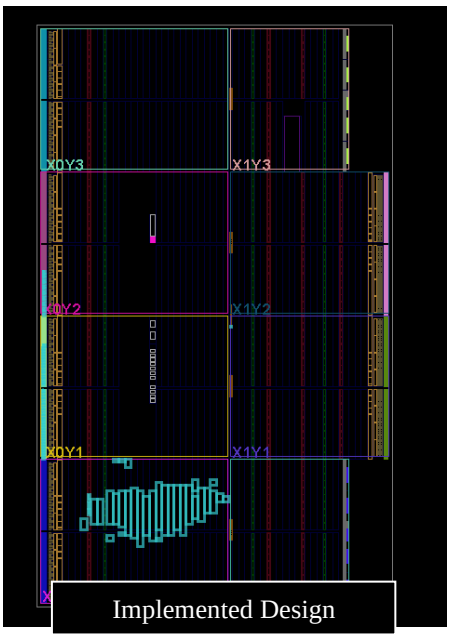
```
1 create_clock -name sysclk -period 32 -waveform {0 16} [get_ports in_clk]
```

Design Timing Summary

| Setup | | Hold | | Pulse Width | |
|---|---|---|---|---|---|
| Worst Negative Slack (WNS): | 0,614 ns | Worst Hold Slack (WHS): | 0,130 ns | Worst Pulse Width Slack (WPWS): | 15,500 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: | 1612 | Total Number of Endpoints: | 1612 | Total Number of Endpoints: | 893 |

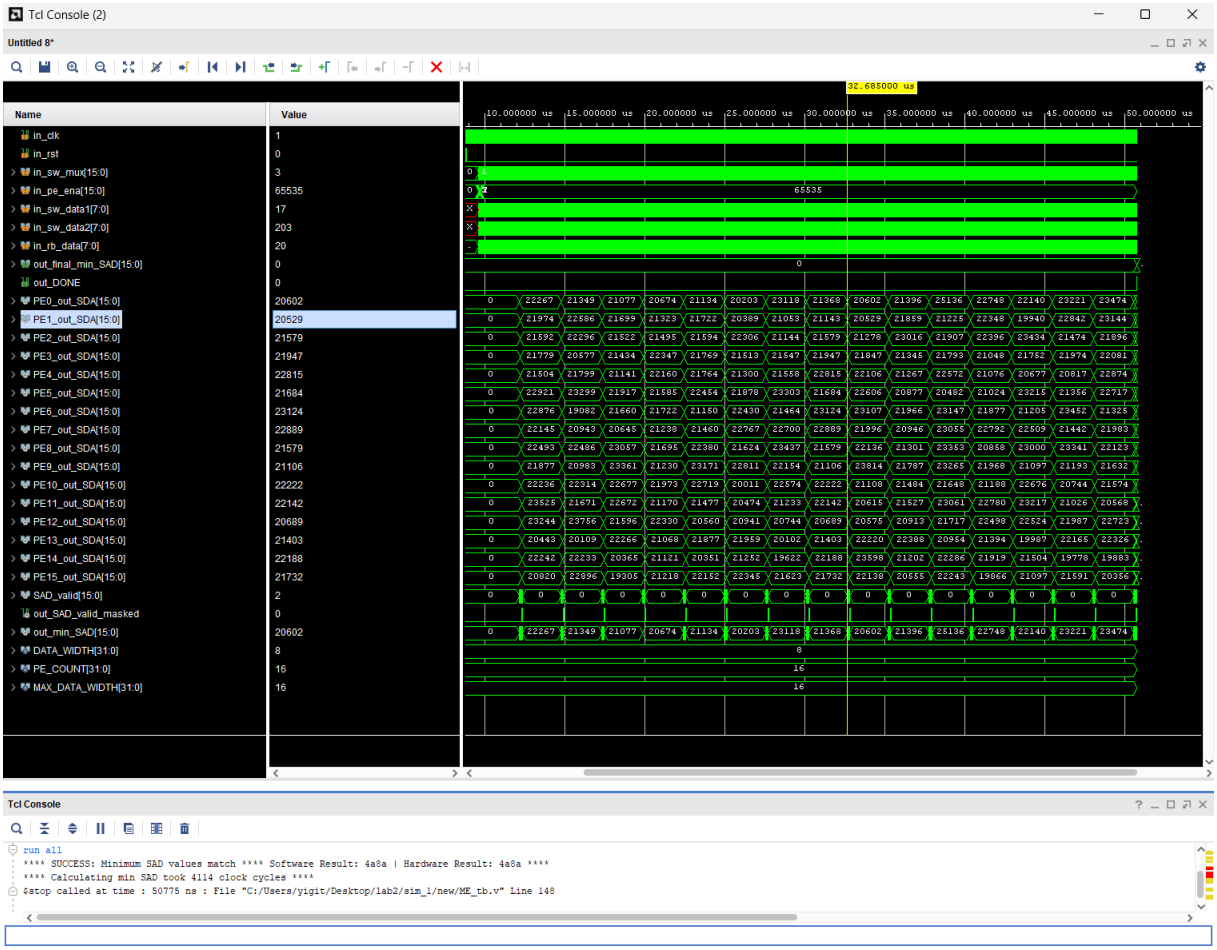
**All user specified timing constraints are met.**

Timing Report

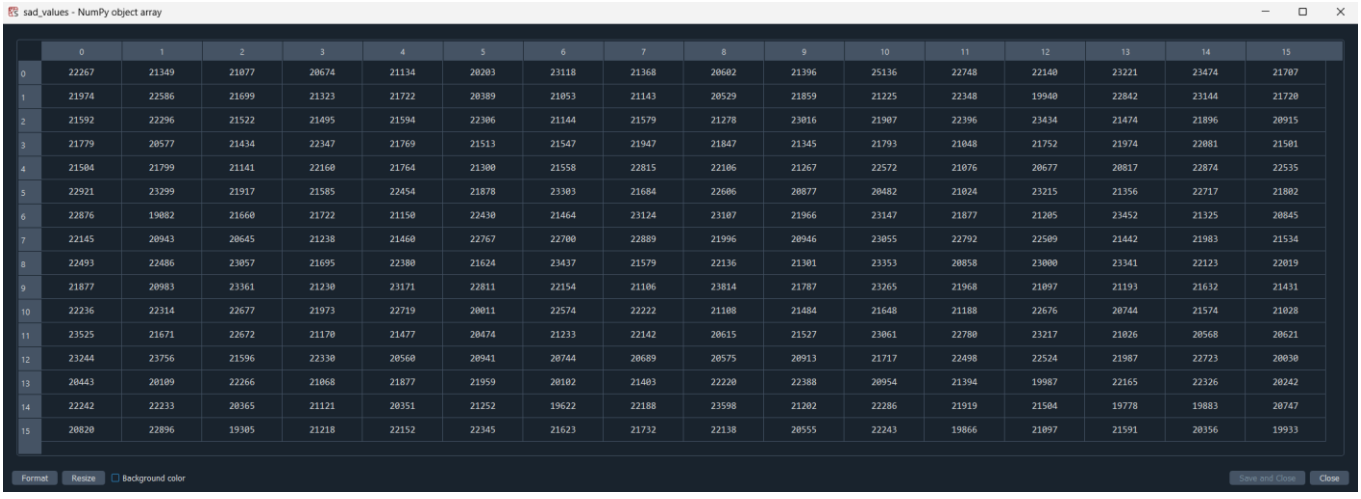




Implemented Design




Simulation and TCL console output




Python SAD Results

SAD results from Python and SAD results generated by PEs in hardware are exactly the same. As seen in the simulation above, the SAD value in the behavioral simulation results gives the same result when compared with the software output. This shows that the circuit is working correctly. In addition, the number of clock cycles required to calculate the minimum SAD of the hardware is printed on the TCL console.