4-Bit Baugh-Wooley Multiplier (IMT2023620 Unnath Ch)

This project implements a 4-bit signed Baugh-Wooley multiplier on a Xilinx FPGA (tested on a Basys 3 board) using the Vivado Suite. It includes two top modules: one for on-chip debugging with VIO/ILA, and another for physical interaction using DIP switches and LEDs.

Github Link

Demonstration & Results

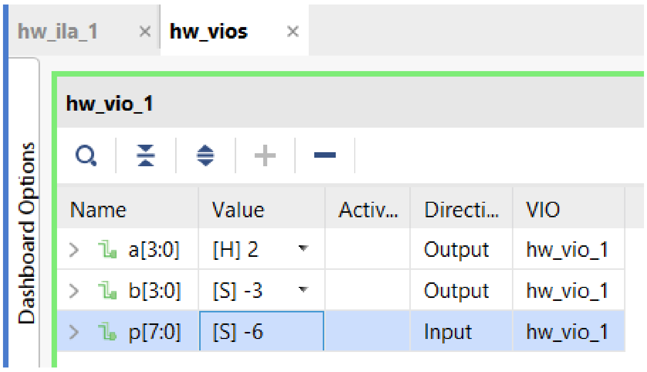
Here are the VIO and ILA dashboards monitoring the multiplier on the FPGA, along with key implementation reports.

VIO Dashboard

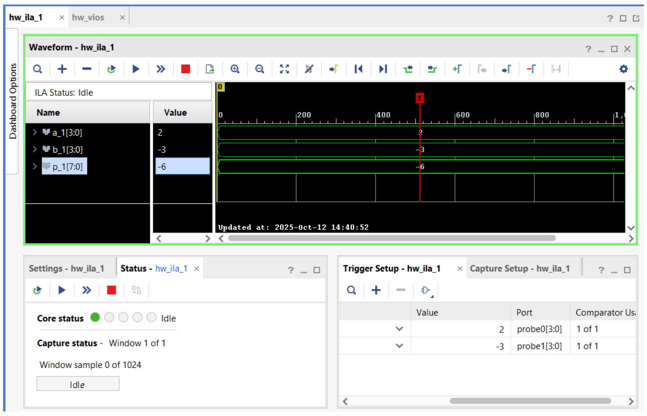
Controlling inputs and viewing the product in real-time.

ILA Waveform

Capturing signal transitions for timing analysis.



Name	Value	Activ...	Directi...	VIO
a[3:0]	[H] 2		Output	hw_vio_1
b[3:0]	[S] -3		Output	hw_vio_1
p[7:0]	[S] -6		Input	hw_vio_1



ILA Status: Idle

Name	Value
a_1[3:0]	2
b_1[3:0]	-3
p_1[7:0]	-6

```
Updated at: 2025-Oct-12 14:40:52
```

Core status Idle

Capture status - Window 1 of 1

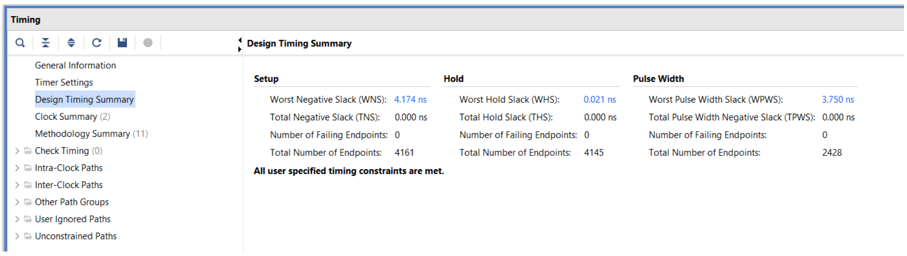
Window sample 0 of 1024

Idle

Value	Port	Comparator Us
2	probe0[3:0]	1 of 1
-3	probe1[3:0]	1 of 1

Timing Summary Report

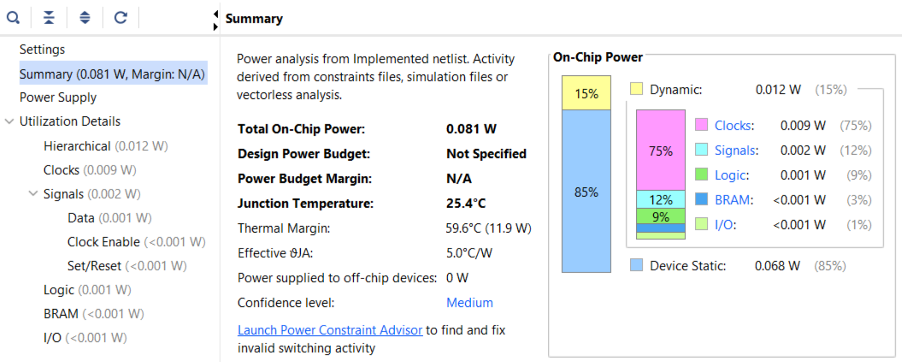
Power Report



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.174 ns	Worst Hold Slack (WHS):	0.021 ns	Worst Pulse Width Slack (WPWS):	3.750 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:	4161	Total Number of Endpoints:	4145	Total Number of Endpoints:	2428

All user specified timing constraints are met.



Summary

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

Total On-Chip Power:	**0.081 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.4°C**
Thermal Margin:	59.6°C (11.9 W)
Effective θJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

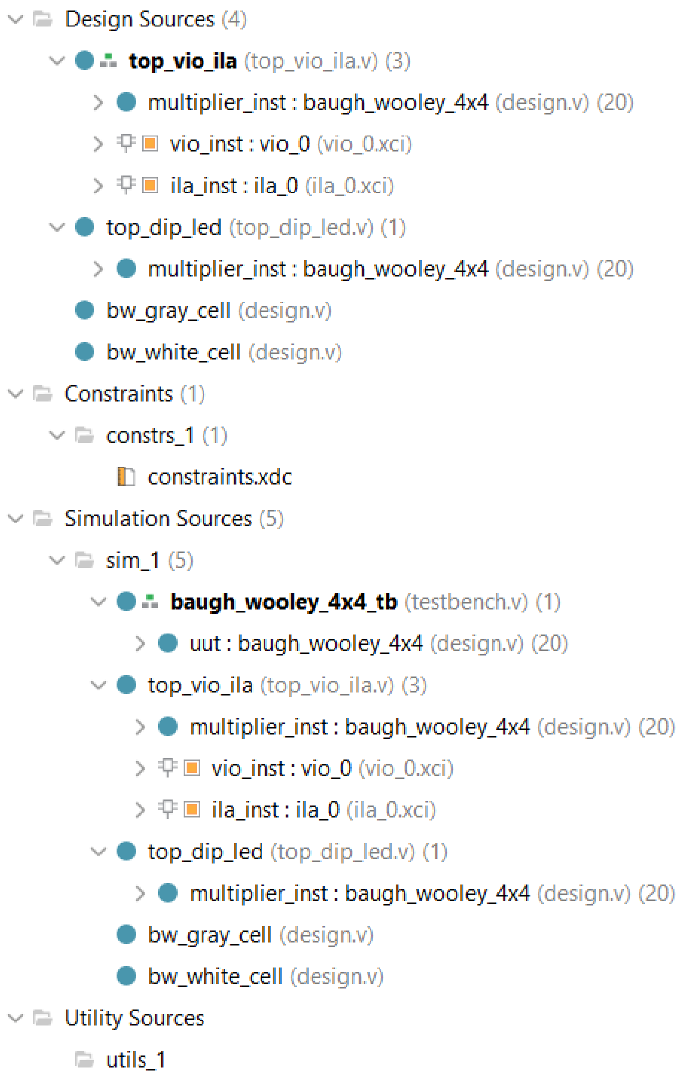
Dynamic:	0.012 W	(15%)
Clocks:	0.009 W	(75%)
Signals:	0.002 W	(12%)
Logic:	0.001 W	(9%)
BRAM:	<0.001 W	(3%)
I/O:	<0.001 W	(1%)
Device Static:	0.068 W	(85%)

Setup for VIO & ILA Debugging 🛠️

This guide explains how to configure the project for on-chip debugging using Vivado's VIO and ILA IP cores.

1. Source File Hierarchy

First, ensure your project sources are set up correctly. The `top_vio_ila` module should be set as the top-level module for this configuration.



```
Design Sources (4)
    top_vio_ila (top_vio_ila.v) (3)
        multiplier_inst : baugh_wooley_4x4 (design.v) (20)
        vio_inst : vio_0 (vio_0.xci)
        ila_inst : ila_0 (ila_0.xci)
    top_dip_led (top_dip_led.v) (1)
        multiplier_inst : baugh_wooley_4x4 (design.v) (20)
    bw_gray_cell (design.v)
    bw_white_cell (design.v)
Constraints (1)
    constrs_1 (1)
        constraints.xdc
Simulation Sources (5)
    sim_1 (5)
        baugh_wooley_4x4_tb (testbench.v) (1)
            uut : baugh_wooley_4x4 (design.v) (20)
        top_vio_ila (top_vio_ila.v) (3)
            multiplier_inst : baugh_wooley_4x4 (design.v) (20)
            vio_inst : vio_0 (vio_0.xci)
            ila_inst : ila_0 (ila_0.xci)
        top_dip_led (top_dip_led.v) (1)
            multiplier_inst : baugh_wooley_4x4 (design.v) (20)
        bw_gray_cell (design.v)
        bw_white_cell (design.v)
Utility Sources
    utils_1
```

2. IP Core Configuration

You'll need to add and configure two IP cores from the IP Catalog as follows:

IP Core	Port	Width	Direction	Purpose
VIO	`probe_in0`	8-bit	Input	Monitors the 8-bit product `p`
	`probe_out0`	4-bit	Output	Controls the 4-bit input `a`
	`probe_out1`	4-bit	Output	Controls the 4-bit input `b`
ILA	`probe0`	4-bit	Input	Captures the signal `a`
	`probe1`	4-bit	Input	Captures the signal `b`

IP Core	Port	Width	Direction	Purpose
	`probe2`	8-bit	Input	Captures the signal `p`

3. Implementation Flow

1. **Create IP Cores**: Follow the detailed steps below to generate the VIO and ILA cores.
2. **Set Top Module**: Right-click `top_vio_ila.v` in the Sources window and select **Set as Top**.
3. **Configure Constraints**: Open `constraints.xdc` and uncomment the clock constraint for your board. **Only the clock constraint should be active.**

   ```
   # For Basys3 board:
   set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
   create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
   ```

4. **Run Full Flow**: Run **Synthesis**, **Implementation**, and **Generate Bitstream**.
5. **Program & Debug**: Open **Hardware Manager**, program the FPGA, and the VIO/ILA dashboards will be available for debugging.

Setup for DIP Switch & LED Usage 💡

If you prefer to use physical switches and LEDs instead of the on-chip debugger:

1. **Set Top Module**: In the Sources window, right-click `top_dip_led.v` and select **Set as Top**.
2. **Update Constraints**: Open `constraints.xdc` and ensure the pin assignments for the **DIP switches** (inputs `a` and `b`) and **LEDs** (output `p`) are uncommented. The VIO/ILA clock constraint should be commented out.
3. **Re-run Flow**: Run **Synthesis**, **Implementation**, and **Generate Bitstream**.
4. **Program FPGA**: Program the device with the new bitstream. You can now control the multiplier with switches and see the result on the LEDs.

Troubleshooting ❓

- **VIO/ILA not in Hardware Manager?** Ensure `top_vio_ila` was set as the top module *before* you generated the bitstream. Reprogram the device.

- **Synthesis fails with "vio_0 not found"?** The VIO IP was not generated correctly. In the Sources window, right-click `vio_0.xci` → **Generate Output Products**.

- **Clock constraint errors?** Make sure you have uncommented **only one** clock constraint in `constraints.xdc` and that the pin assignment (`W5`, `E3`, etc.) matches your specific FPGA board.
