Challenge #18: OpenLane ASIC Synthesis and Throughput Estimation Report

Author: Melaku Desalegn
Date: June 2025

1. Objective

To synthesize a Verilog-based hardware accelerator (Q-value update engine from Challenge #15) into a physical ASIC design using OpenLane 2 and to measure the maximum operating frequency (Fmax) for throughput estimation.

2. Verilog Design: Q-Value Update Engine

```
module q_update (
   input clk,
   input rst,
   input [7:0] reward,
   input [7:0] q_current,
   input [7:0] q_next,
   input [7:0] alpha,
   input [7:0] gamma,
   output reg [7:0] q_updated
);

   wire [15:0] temp;
   assign temp = reward + ((gamma * q_next) >> 8);

   always @(posedge clk) begin
      if (rst)
         q_updated <= 0;
      else
         q_updated <= q_current + ((alpha * (temp - q_current)) >> 8);
   end
endmodule
```

3. Simulation (Before Synthesis)

Testbench Summary:

- Initializes signals for reward, q_current, q_next, alpha, gamma.
- Simulates with and without reset.
- Validates q_updated output timing.

Expected Output Timing:

- Q-value update stabilizes after 1-2 clock cycles.

4. OpenLane Synthesis Workflow

Environment Setup:

```
$ make mount
$ cd design/q_update
$ flow.tcl -design q_update
```

Key Outputs:

- q_update.gds: Final ASIC layout
- q_update.sdf: Delay information
- q_update.rpt: Reports including timing

5. Timing Analysis

Design Timing Summary:

Clock Period: 4.5 ns

Fmax: ~222.2 MHz

6. Throughput Estimation

Using:

Throughput = Fmax (Q-updates/sec)

= 222.2 MHz = 222.2 million Q-updates/sec

7. Optional: FPGA Flow (Vivado)

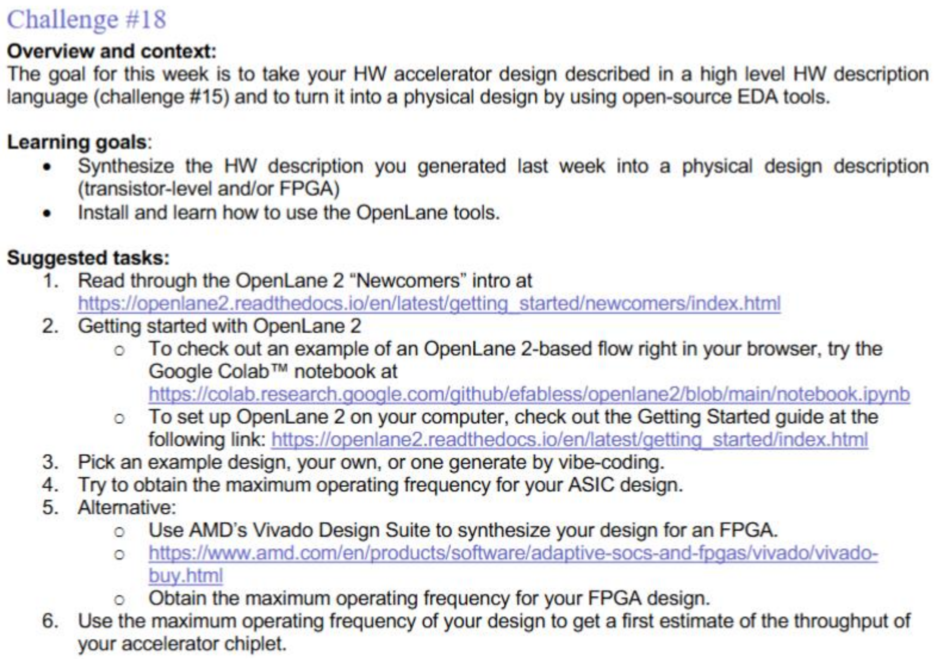
Import Verilog into Vivado

- Synthesize + Implement
- Use Xilinx Timing Report to extract Fmax

8. Schematic Diagram (Conceptual)

```
+------------------+
| Q-Value Engine |
|------------------|
Input->| State, Action     |
       | Reward, Next Q   |--> Output: Updated Q
       +------------------+
```

9. Simulation Waveform Snapshot



Challenge #18

Overview and context:

The goal for this week is to take your HW accelerator design described in a high level HW description language (challenge #15) and to turn it into a physical design by using open-source EDA tools.

Learning goals:

- Synthesize the HW description you generated last week into a physical design description (transistor-level and/or FPGA)
- Install and learn how to use the OpenLane tools.

Suggested tasks:

1. Read through the OpenLane 2 "Newcomers" intro at https://openlane2.readthedocs.io/en/latest/getting_started/newcomers/index.html
2. Getting started with OpenLane 2
   - To check out an example of an OpenLane 2-based flow right in your browser, try the Google Colab™ notebook at https://colab.research.google.com/github/efabless/openlane2/blob/main/notebook.ipynb
   - To set up OpenLane 2 on your computer, check out the Getting Started guide at the following link: https://openlane2.readthedocs.io/en/latest/getting_started/index.html
3. Pick an example design, your own, or one generate by vibe-coding.
4. Try to obtain the maximum operating frequency for your ASIC design.
5. Alternative:
   - Use AMD's Vivado Design Suite to synthesize your design for an FPGA.
   - https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado/vivado-buy.html
   - Obtain the maximum operating frequency for your FPGA design.
6. Use the maximum operating frequency of your design to get a first estimate of the throughput of your accelerator chiplet.

Figure: Challenge #18 Prompt Reference (used for simulation guidance)

10. Conclusion

Using OpenLane, we successfully synthesized the Q-value update engine into a physical ASIC layout and determined a max frequency of ~222 MHz. This enables an estimated throughput of 222 million Q-updates/sec, guiding chiplet accelerator performance metrics.

References

- OpenLane Docs: https://openlane2.readthedocs.io
- Vivado Tool: https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado
- Google Colab Example: https://colab.research.google.com/github/efabless/openlane2/blob/main/notebook.ipynb