Flash Attention on FPGA

Aakarsh A
Department of Computer Science and Automation
Indian Institute of Science

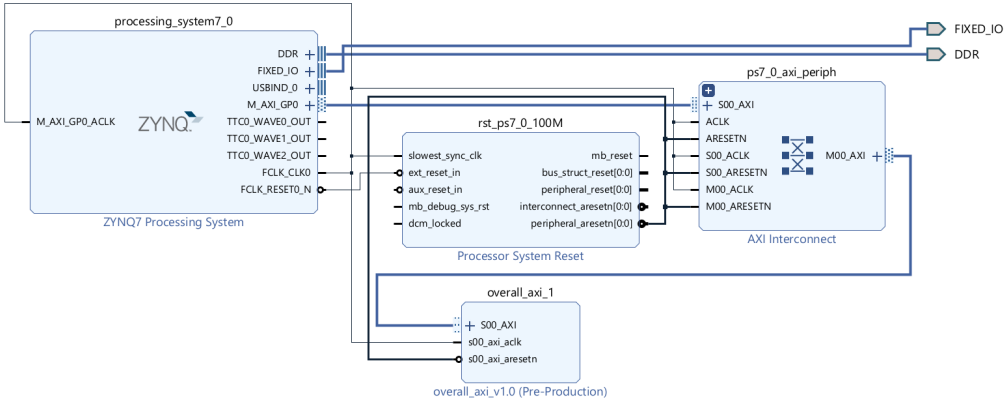
June 13, 2025

Overview

1. Implementation results from Zedboard
2. Phase-2 timing performance improvement
3. Future Work

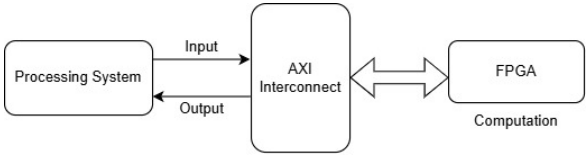
Implementation Result: 1

PS & PL Approach



Block Design integrating PS and PL via AXI Interconnect, and our designed RTL block is created as another AXI-4 Lite block to effectively interface with PS.

Implementation Result: 2



Implementation on Zedboard

```
------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints     WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total Endpoints     WPWS(ns)     TPWS(ns)
    -------      -------  ---------------------  -------------------     -------      -------  ---------------------  -------------------     --------     --------
      0.175        0.000                      0                 1430       0.062        0.000                      0                 1430        1.353        0.000


All user specified timing constraints are met.


------------------------------------------------------------------------------------------
| Clock Summary
| -------------
------------------------------------------------------------------------------------------


Clock        Waveform(ns)         Period(ns)      Frequency(MHz)
-----        ------------         ----------      --------------
clk_fpga_0   {0.000 2.333}        4.666           214.316
```

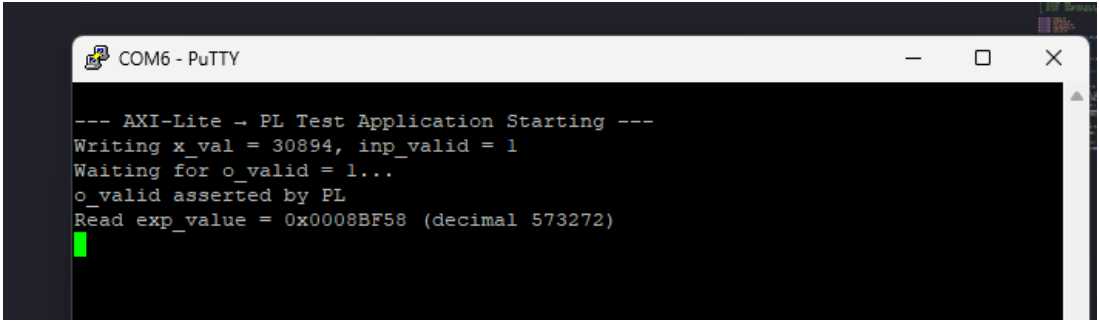
Timing info for negative exponential after frequency optimization(Achieved 214 MHz). Further increment possible in Alveo V80 based on their DSPs' processing speed.

Xilinx SDK

```
xil_printf(ctrl1: "Writing x_val = %d, inp_valid = %d\r\n", x_val, inp_valid);
Xil_Out32(Addr: AXI_ADDRESS + 0x00, Value: x_val);
Xil_Out32(Addr: AXI_ADDRESS + 0x04, Value: inp_valid);

xil_printf(ctrl1: "Waiting for o_valid = 1...\r\n");
do {
    read_data = Xil_In32(Addr: AXI_ADDRESS + 0x0C);  // read entire 32-bit reg
    out_valid   = read_data & 0x1;                   // extract bit[0]
} while (out_valid == 0);

// At this point, o_valid == 1
xil_printf(ctrl1: "o_valid asserted by PL\r\n");

y = Xil_In32(Addr: AXI_ADDRESS + 0x08);

// while (1) {
// Might add extra software logic.
// }
xil_printf(ctrl1: "Read exp_value = 0x%08x (decimal %d)\r\n", (unsigned) y, (int) y);
```

Generate Bitstream, export Hardware, and set it as Platform hardware in Vitis SDK, and Code up the application/task as needed.

Complete PS-PL flow ouput



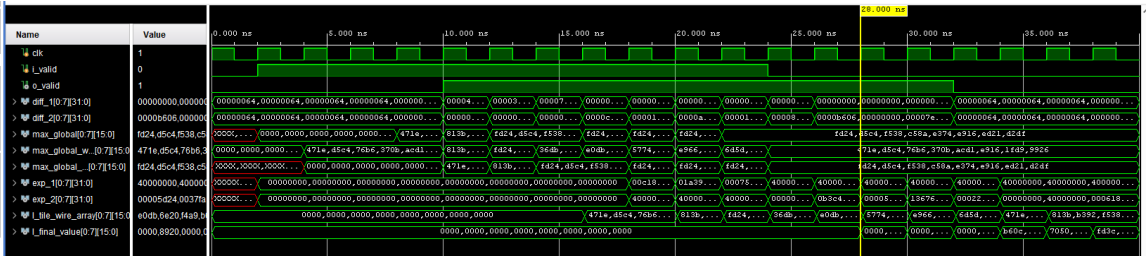
The output was captured on PuTTY from the Zedboard through UART. For further testing, the inputs can be sent one by one for pipelined output.

Initial Phase-2 design timing analysis

```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total Endpoints     WPWS(ns)
    -------      -------  ---------------------  -------------------      -------      -------  ---------------------  -------------------     --------
     -2.354     -236.053                    132                 1021        0.122        0.000                      0                 1021        4.500


Timing constraints are not met.


------------------------------------------------------------------------------------------------
| Clock Summary
| -------------
------------------------------------------------------------------------------------------------

Clock    Waveform(ns)       Period(ns)      Frequency(MHz)
-----    ------------       ----------      --------------
sys_clk  {0.000 5.000}      10.000          100.000
```

- The design had a latency of 7 clock cycles, but due to quick BRAM access and a non-pipelined multiplier, the maximum operating frequency was less than 100 MHz at 81 MHz.

Improved Phase-2 design: 1



- The design has a latency of 13 CC, compared to the previous design of 7 CC.
- BRAM access: 2 CC, comparison and storage: 2 CC, subtraction: 1 CC, e^{-x}: 3 CC, Pipelined Multiplication: 3 CC and addition: 1 CC. Total = 13 CC latency. After the first output, every other output is obtained in consecutive cycles.
- Further improvements: Improve BRAM access, and Multiplier's efficiency - dependent on Alveo V80.

Improved Phase-2 design: 2

```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total Endpoints     WPWS(ns)     TPWS(ns)
    -------      -------  ---------------------  -------------------      -------      -------  ---------------------  -------------------     --------     --------
      0.067        0.000                      0                  464        0.107        0.000                      0                  464        2.276        0.000


All user specified timing constraints are met.


------------------------------------------------------------------------------------------------
| Clock Summary
| -------------
------------------------------------------------------------------------------------------------

Clock    Waveform(ns)         Period(ns)      Frequency(MHz)
-----    ------------         ----------      --------------
sys_clk  {0.000 3.080}        6.160           162.338
```

- Though it had a latency of 13 CC, the maximum achievable frequency increased to 162 MHz from 81 MHz. (Might further increase with addition of PS-PL interface).
- The increment percentage might not directly correlate with Alveo's performance, but will definitely improve the timing performance overall by a significant margin.

Future Work

1. Resolve issues with Phase-2 Zedboard Implementation.
2. RTL implementation of Phase-3 with integration with Phase-2 with implementation.
3. Verification of Phase-2 on the Alveo V80 FPGA.
4. More info on NoC configuration on V80.

THE END

Feedback & Improvement Ideas?