



Lab1 Intro

Vitis HLS Design Flow

Objectives

- ▶ After completing this lab, you will be able to:
 - Create a project in Vitis HLS
 - Run C-simulation
 - Use debugger
 - Synthesize and implement the design using the default options
 - Use design analysis perspective to see what is going on under the hood
 - Understand and analyze the generated output

The Design

- ▶ This lab uses a simple matrix multiplication example to walk you through the Vitis HLS project creation and analysis steps. The design consists of three nested loops. The Product loop is the inner most loop performing the actual elements product. The Col loop is the outer-loop which feeds next column element data with the passed row element data to the Product loop. Finally, Row is the outer-most loop. The `res[i][j]=0` (line 79) resets the result every time a new row element is passed and new column element is used

```
67 #include "matrixmul.h"
68
69 void matrixmul(
70     mat_a_t a[MAT_A_ROWS][MAT_A_COLS],
71     mat_b_t b[MAT_B_ROWS][MAT_B_COLS],
72     result_t res[MAT_A_ROWS][MAT_B_COLS])
73 {
74     // Iterate over the rows of the A matrix
75     Row: for(int i = 0; i < MAT_A_ROWS; i++) {
76         // Iterate over the columns of the B matrix
77         Col: for(int j = 0; j < MAT_B_COLS; j++) {
78             // Do the inner product of a row of A and col of B
79             res[i][j] = 0;
80             Product: for(int k = 0; k < MAT_B_ROWS; k++) {
81                 res[i][j] += a[i][k] * b[k][j];
82             }
83         }
84     }
85 }
```

Procedure

- ▶ Create a project after starting Vitis HLS in GUI mode
- ▶ Run C simulation
 - to understand the design behavior
- ▶ Run the debugger
 - to see how the top-level module works
- ▶ Synthesize the design
- ▶ Analyze the generated output using the Analysis perspective
- ▶ Run C/RTL co-simulation
 - to perform RTL simulation
- ▶ View simulation results in Vitis
 - to understand the IO protocol
- ▶ Export RTL in the Evaluate mode and run the implementation

Summary

- ▶ Completed the major steps of the high-level synthesis design flow using Vitis HLS.
- ▶ Created a project, added source files, synthesized the design, simulated the design, and implemented the design.
- ▶ Learned about how to use the Analysis perspective to understand the scheduling.



Thank You

Disclaimer and Attribution

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

© Copyright 2021 Advanced Micro Devices, Inc. All rights reserved. Xilinx, the Xilinx logo, AMD, the AMD Arrow logo, Alveo, Artix, Kintex, Kria, Spartan, Versal, Vitis, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

