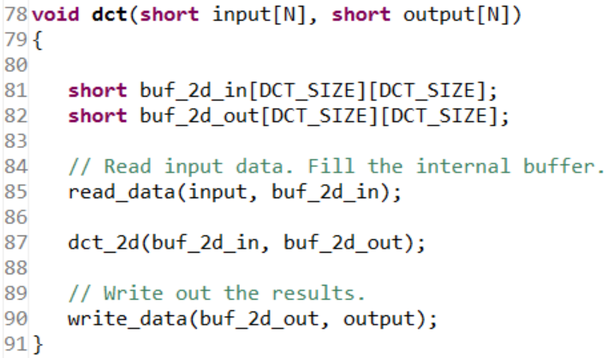
Lab3 Intro
Improving Area and Resources

Objectives

- After completing this lab, you will be able to:
  - Manage BRAM and DSP48 resource utilization
  - Improve memory bandwidth
  - Balance resource utilization and performance
  - Distinguish between DATAFLOW directive and Configuration Command functionality

The Design

- The design under consideration is a Discrete Cosine Transformation (DCT) function on a 8x8 block of data
  - The top-level function dct implements 2D DCT algorithm by first processing each row of the input array via a 1D DCT then processing the columns of the resulting array through the same 1D DCT. It calls read_data, dct_2d, and write_data functions.
  - The read_data function consists of two loops – RD_Loop_Row and RD_Loop_Col.
  - The write_data function is defined consists of two loops to perform writing the result.



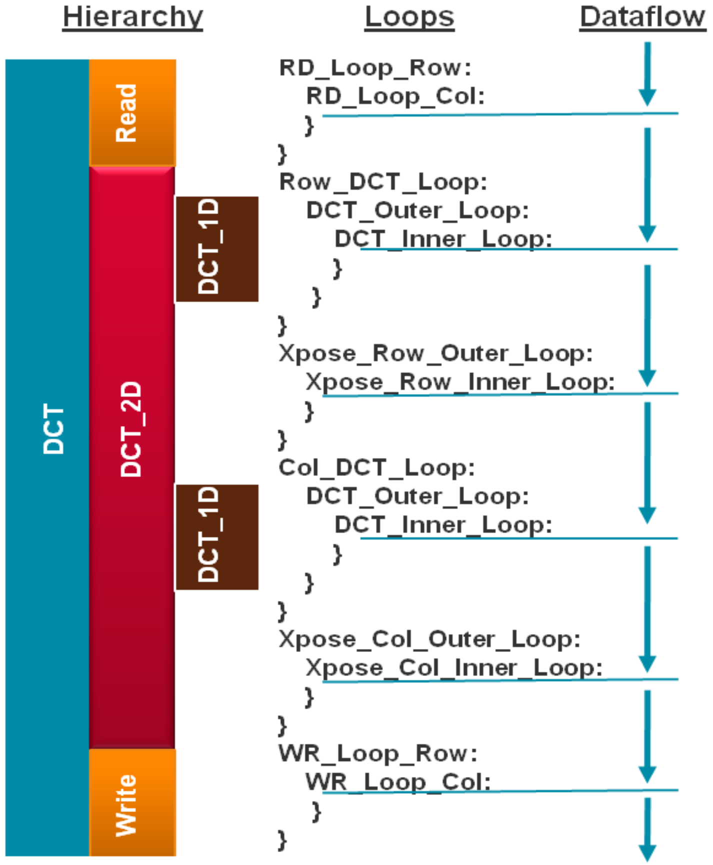
```
void dct(short input[N], short output[N])
{

   short buf_2d_in[DCT_SIZE][DCT_SIZE];
   short buf_2d_out[DCT_SIZE][DCT_SIZE];

   // Read input data. Fill the internal buffer.
   read_data(input, buf_2d_in);

   dct_2d(buf_2d_in, buf_2d_out);

   // Write out the results.
   write_data(buf_2d_out, output);
}
```



```
RD_Loop_Row:
  RD_Loop_Col:
  }
}
Row_DCT_Loop:
  DCT_Outer_Loop:
    DCT_Inner_Loop:
    }
  }
}
Xpose_Row_Outer_Loop:
  Xpose_Row_Inner_Loop:
  }
}
Col_DCT_Loop:
  DCT_Outer_Loop:
    DCT_Inner_Loop:
    }
  }
}
Xpose_Col_Outer_Loop:
  Xpose_Col_Inner_Loop:
  }
}
WR_Loop_Row:
  WR_Loop_Col:
  }
}
```

Procedure

- Compile the design in command mode and perform C-verification
- Open the project in Vitis HLS GUI, synthesize, and review results
- Simulate the design
- Improve performance using pipeline
- Optimize fine-grained parallelism
- Improve memory bandwidth
- Apply DATAFLOW directive to improve performance
- Apply RESHAPE directive and analyze

Summary

- In this lab, you learned various techniques to improve the performance and balance resource utilization.
- PIPELINE directive when applied to outer loop will automatically cause the inner loop to unroll. When a loop is unrolled, resources utilization increases as operations are done concurrently. Partitioning memory may improve performance but will increase BRAM utilization.
- When INLINE directive is applied to a function, the lower level hierarchy is automatically dissolved. When DATAFLOW directive is applied, the default memory buffers (of ping-pong type) are automatically inserted between the top-level functions and loops.
- The RESHAPE directive will allow multiple accesses to BRAM, however, care should be taken if a single element requires modification as it will result in read-modify-write operation for the entire word. The console logs can provide insight on what is going on.

AMD
XILINX

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

AMD
XILINX