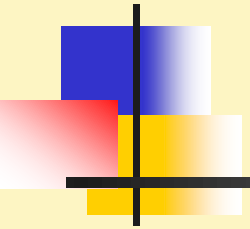


HLS Assignment

Image Convolution
with
Vivado HLS

Before starting the actual HLS Assignment

- Vivado HLS Welcome Page
- You can find from here all the needed tutorials and documentation
- You can start from here learning the Vivado HLS tool
- You can start from here creating your new project

Before starting the actual HLS Assignment

- If you did not do it yet, first download, get licenses for, and learn usage of **Vivado HLS** and **Vivado** (you need them both), and understand the produced information, when using the earlier specified information from HLS Lecture 1
- **New Vivado 2018.3 release**:
  - in December 2017 Xilinx made available the new Vivado Design Suite HLx Editions present in Vivado 2018.3 and including **Vivado HLS** and **Vivado HL WebPACK Edition**:

    http://www.xilinx.com/products/design-tools/vivado.html

    http://www.xilinx.com/support/download.html

  - The new HL WebPACK Edition also includes a limited version of Vivado HLS (a. o. not all FPGAs are supported)

Before starting the actual HLS Assignment

- If you did not do it yet, read carefully Chapter 2: High-level Synthesis Introduction from Vivado Design Suit Tutorial: High-Level Synthesis
- Watch carefully Vivado HLS Getting Started Video Tutorial
- Perform the design from the tutorial in the Vivado HLS tools, just follow all the steps in the tutorial
- Analyze and understand the HLS results, all performance estimates, schedules and other reports generated in Synthesis and Analysis Perspectives, etc.

HLS Assignment: Attached files

- The attached files for this HLS assignment contain:
  - convolution5x5: the original version of the image convolution function in C - NOT synthesizable due to use of not fixed size arrays
  - img_conv_5x5.c: synthesizable version of the convolution function of one row of the image inptr[ ]
  - img.conv.5x5.h: header file containing all needed definitions as recommended by HLS guidelines
- Preserve all files of your project: you will have to use them for presentation and explanation of your design to me

img_conv_5x5.c

```
#include "img_conv_5x5.h"

void img_conv_5x5(unsigned char inptr[X_N*Y_N],
                  unsigned char outptr[X_N],
                 // int8 x_dim,
                  char mask[M_N],
                  int  shift)
{
   unsigned char     *IN1,*IN2,*IN3,*IN4,*IN5;
   unsigned char     *OUT;
   short    pix10,  pix20,  pix30,  pix40,  pix50;
   short    mask10, mask20, mask30, mask40, mask50;
   int      sum,    sum00,  sum11, sum22, sum33, sum44;
   int      i;
   int      j;

// seting pointers to read data from five rows of inptr[]
// and to write the results to a row OUT
   IN1      =   inptr;
   IN2      =   IN1 + X_N;  // X_N row width
   IN3      =   IN2 + X_N;
   IN4      =   IN3 + X_N;
   IN5      =   IN4 + X_N;
   OUT      =   outptr;
```

img_conv_5x5.c 2

```
// go through the row of inptr[] image with j index
// produce one convolution output OUT pixel per iteration

for (j = 0; j < X_N - 5 ; j++)
    {
       sum = 0;

       for (i = 0; i < 5; i++)
// calculate convolution sum by summing mask*pixel
       {
          pix10  =   IN1[i];
          pix20  =   IN2[i];
          pix30  =   IN3[i];
                       pix40  =   IN4[i];
                       pix50  =   IN5[i];

          mask10 =   mask[i];
          mask20 =   mask[i + 5];
          mask30 =   mask[i + 10];
                     mask40 =   mask[i + 15];
                     mask50 =   mask[i + 20];

          sum00  =   pix10 * mask10;
          sum11  =   pix20 * mask20;
          sum22  =   pix30 * mask30;
                     sum33  =   pix40 * mask40;
                     sum44  =   pix50 * mask50;

          sum   +=   sum00 + sum11 + sum22 + sum33 + sum44;
       }
```

img_conv_5x5.c 3

```
// move to next column of image
        IN1++;
        IN2++;
        IN3++;
                        IN4++;
        IN5++;

// calculate OUT pixel
        sum = (sum >> shift);

        if ( sum <  0  )       sum = 0;
        if ( sum > 255 )       sum = 255;

        *OUT++   =       sum;
    }

} //end func
```

img_conv_5x5.h

```
/*********************************************************
```

This file defines types and constants for use in img_conv_5x5 convolution function.

Types defined here are NOT used in the function but show how it should be done according to VivadoHLS guidelines.

```
*********************************************************/
#ifndef _IMG_CONV_5X5_
#define _IMG_CONV_5X5_

#include <stdio.h>
#include "ap_cint.h"

#define X_N 64  // width of row
#define Y_N 32  // number of rows
#define M_N 25  // size of mask array

typedef const unsigned char  imgpix;
typedef unsigned char  convpix;
typedef const char mpix;
// arbitary precision types
typedef int3 i_t;  // limited rage to save hardware
typedef int10 j_t; // limited rage to save hardware

void img_conv_5x5(unsigned char inptr[X_N*Y_N],
                  unsigned char outptr[X_N],
                 // int8 x_dim,
                  char mask[M_N],
                  int  shift) ;
#endif
```

Minimum HLS Assignment

- Start Vivado HLS and Create New Project using the attached source files img_conv_5x5.c and img.conv.5x5.h
- Specify img_conv_5x5 as Top Function
- Initially do not add C-based testbench files, but click Next
- Perform Solution Configuration as follows:
- Select Clock Period to be 10 ns
- Select Part to be xa7z030fbv484-1i
- Run C Synthesis using an appropriate Toolbar Button.
- Analyze and understand the Synthesis Reports (Performance Estimates and Utilization Estimates) in details

Minimum HLS Assignment

- Switch to Analysis Perspective: Analyze and understand (in details) the schedule in Performance View, as well as, the schedule and resources in Resource View
- To directly get from Vivado HLS the logic-level scheme, physical implementation and related reports for your RTL design obtained from C Synthesis perform the following:
  - Export RTL using Solution Toolbar Button: Export RTL
  - In the dialog box select in Format Selection: IP Catalog; in Evaluate Generated RTL: Verilog; and select both: Vivado RTL Synthesis and Place and Route
  - Vivado HLS will then export RTL and use Vivado to perform the logic and physical synthesis without showing the process and results to you

Minimum HLS Assignment

- After several minutes of running Vivado from Vivado HLS you will see in the Vivado HLS Console: **Finished export RTL**, and in the Vivado HLS Explorer Pane (left window): in ‘solution1’->’impl’->’verilog’-> **file *project.xpr* will emerge**
- After this will happen start Vivado, in Vivado go to directory solution1/impl/verilog and open project ***project***
- **You are now in Vivado** (not in Vivado HLS) and you have access to your whole RTL, logic and physical design project, including the logic-level and physical-level implementations and related reports

- Open Synthesized Design in the Flow Navigator Pane (left window) to see and analyze Schematic (logic-level scheme) and related reports
- Try to find and understand relationships (differences and similarities) between the values of estimates in the reports after C synthesis and after the implementation

Minimum HLS Assignment

- Read carefully Chapter 3: C Validation from Vivado Design Suit Tutorial: High-Level Synthesis
- Watch carefully Verifying your Vivado High-level Synthesis (HLS) Design
- Prepare C-based testbench files for your img_conv_5x5 design
- Verify your img_conv_5x5 HLS design both at the C level and RTL level
- Analyze and understand the verification results and reports in various perspectives

Extended HLS Assignment

- Additional work!
- img_conv_5x5.c - computes the convolution function for one row of the image inptr[ ]
- Write in C a main function whole_img_conv_5x5.c that computes the convolution function for the whole image inptr[ ], when iteratively calling img_conv_5x5
- Run C Synthesis of the convolution of the whole image
- Analyze and understand the related reports
- Precise description of the C coding styles and limitations for Vivado HLS can be found in the file “xilinx-ug902-vivado-high-level-synthesis.pdf“, Chapter 3: High-level Synthesis Coding Styles

HLS Assignment: Evaluation

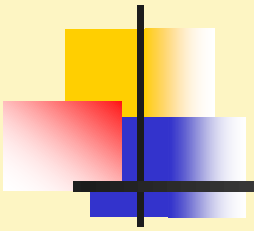
- Assignment: High-level synthesis – 20% of the whole course
- Minimum HLS Assignment: **0 to 16 points**
- Minimum + Extended HLS Assignment: **0 to 20 points**
- The number of points depends on:
  - How much you did
  - How good you understand and can explain what you did
- Extended HLS Assignment is not compulsory, but may result in additional points (within max. 20 points)
- <=3 students/group: every student has to make it on own laptop – **to be evaluated come with your laptop!**
- Preserve all files of your project: you will have to use them for presentation and explanation of your design to me

HLS Assignment: Evaluation

- Deadline: **11.03.2018**
- If your group is earlier ready and you want to be earlier evaluated and examined, please inform me on this per e-mail, when clearly specifying your group number
- Evaluation is based on the work performed and assignment results presentation by individual students of the group
  Prepare well to present the assignment results efficiently:
  - Result presentation: max. 15 min. per group
- Assignment results presentation has to be based on files of your project, but if you consider it important, you may prepare a max. 5 min PowerPoint presentation to better present the assignment results

HLS Exam Questions

- Exam will check if the student understands the HLS problems, their basic solutions and basic solution algorithms, and can apply the solutions and algorithms to solve the problems
- Questions to be answered by individual students:
  - Approximately 10 - 15 min. per person
- Two kinds of questions:
  - questions related to the assignment results to be answered based on files of your project
  - questions related to scheduling and binding/resource reuse
- The second kind questions will check if student understands the scheduling and resource binding/reuse ideas and algorithms discussed during the lecture, and can apply them to schedule and bind a CDFG according to the specified requirements



HLS Exam Questions

- To prepare to answering the questions related to scheduling and binding:
  - each student has to solve the HLS questions specified in the document: HLS Assignment Part 2 - Exam Preparation
  - each student has to take with him/her the printed/paper version of this document including his/her solutions of the HLS questions to his/her HLS exam

HLS Assignment Evaluation and Exam

- Individual Assignment Evaluation and HLS Exam will be scheduled per HLS group between 12.03 and 22.03
- Information on the time-slots in which I will be available to evaluate and examine you will be made available after 18.02 in Canvas
- Every HLS group is asked to sign for the HLS Evaluation and Exam to one of the available time-slots by 25.02 at the latest
- If your group is earlier ready and you want to be earlier evaluated and examined, please inform me on this per e-mail (L.Jozwiak@tue.nl), while clearly specifying your group number and when you can be evaluated and examined.