OpenCV acceleration battle:

OpenCL on Firefly-RK3288(MALI-T764) vs. FPGA on ZedBoard(Zynq-7020)

Noritsuna Imamura

noritusna@siprop.org

SIProp

Agenda

SIProp

- OpenCV for OpenCL
- OpenCV for FPGA

SIProp

!!!!!!ATTENTION!!!!!!

SIProp

- This Slide is NOT
  - OpenCL for GPU vs. FPGA

SIProp

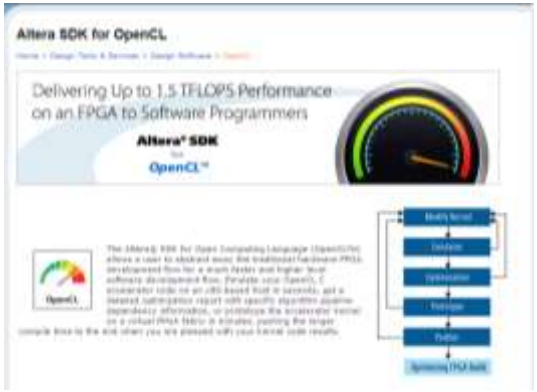
- OpenCL for FPGA
  - Today's Agenda
    - OpenCV for OpenCL
    - OpenCV for FPGA

OpenCL SDK for FPGA

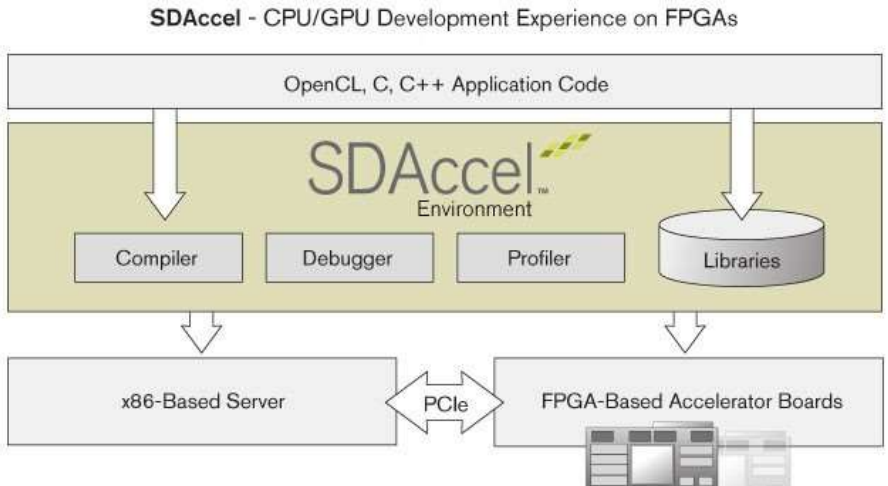
SIProp

SIProp

- Altera SDK for OpenCL
  - http://www.altera.com/products/software/opencl/opencl-index.html



- Xilinx SDAccel
  - http://www.xilinx.com/products/design-tools/sdx/sdaccel.htm



Advantage of FPGA

SIProp

- Direct connect Peripherals to FPGA.
  - GPGPU must bypass CPU/Memory bus.
- Ex. Network peripheral

About OpenCL for FPGA

SIProp

- Programming Language for FPGA
  - OpenCL Compiler for RTL(Register Transfer Level)
  - OpenCL Runtime Library
- Why for usage?
  - For “Software Engineers”
  - Easy to Program for FPGA

SIProp

Advantage of OpenCL for FPGA

SIProp

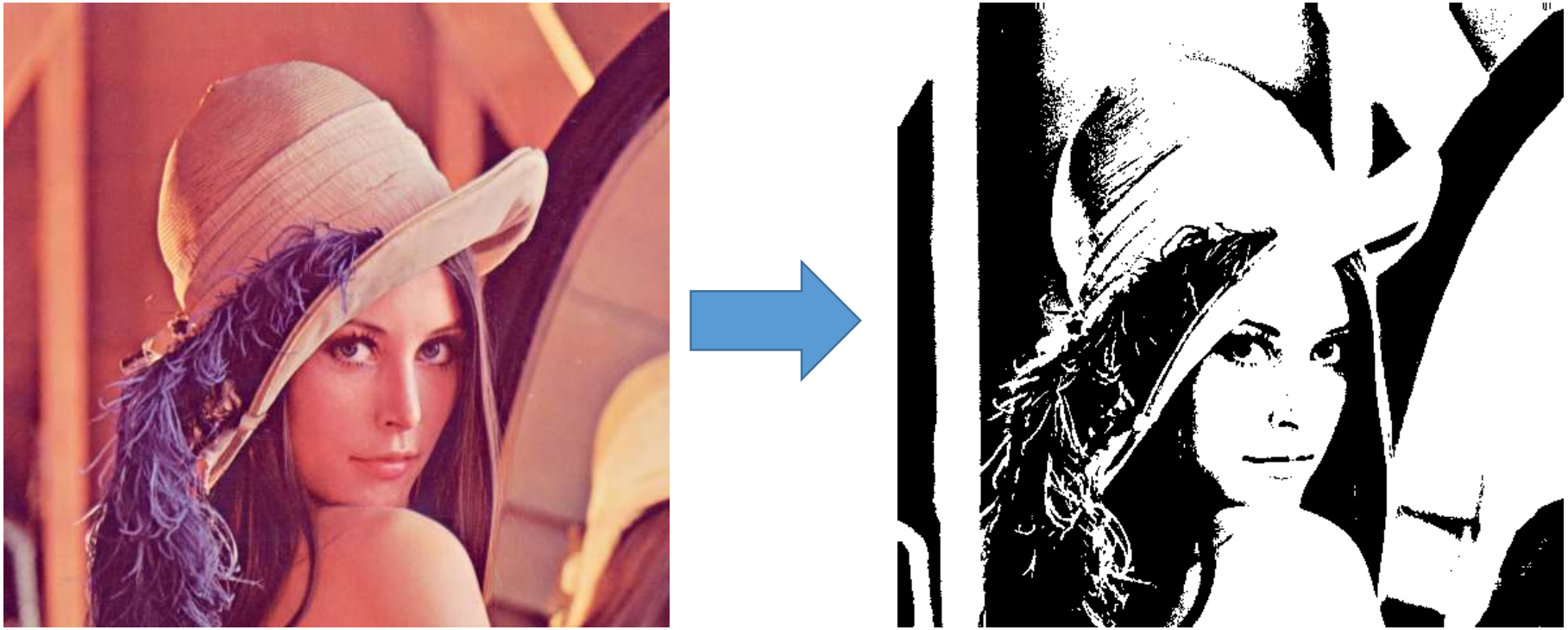
- High-Level Synthesis
  - C/C++/Java/Python for HDL
- FPGA features
  - No Memory
    - FPGA has SRAM/SDRAM. But small size & big overhead.
  - Parallel Processing
    - Input number is Parallel number.

Streaming Processing for No Memory

SIProp

SIProp

- Ex. Effect(pixel by pixel)

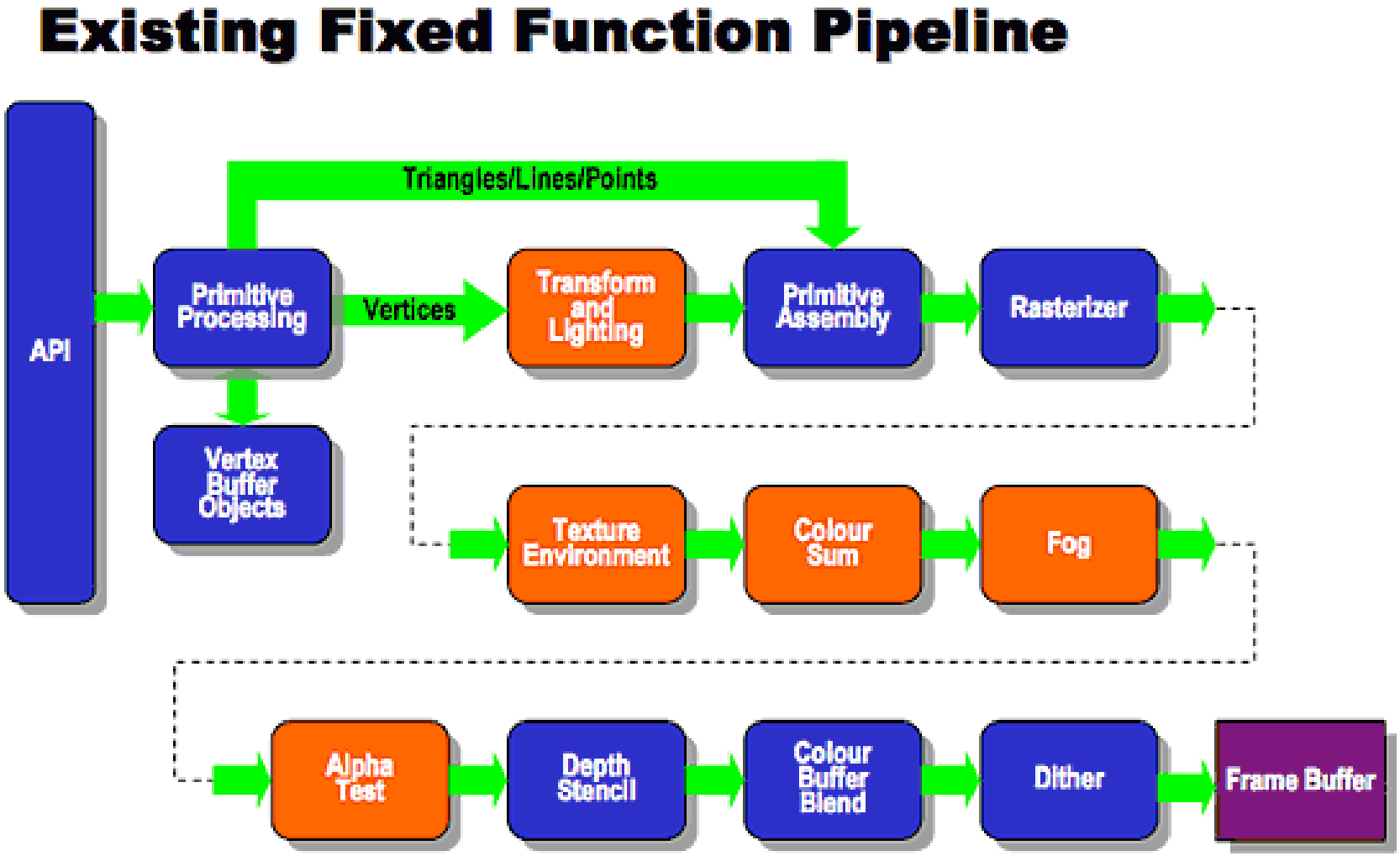


Pipeline Processing for Parallel

SIProp

SIProp

- Ex. OpenGL Architecture



- OpenCV for OpenCL

About OpenCL for OpenCV 1/2

SIProp

- OpenCL Functions for OpenCV
  - cv::ocl::xxx
    - Wrapped Functions as OpenCV Functions

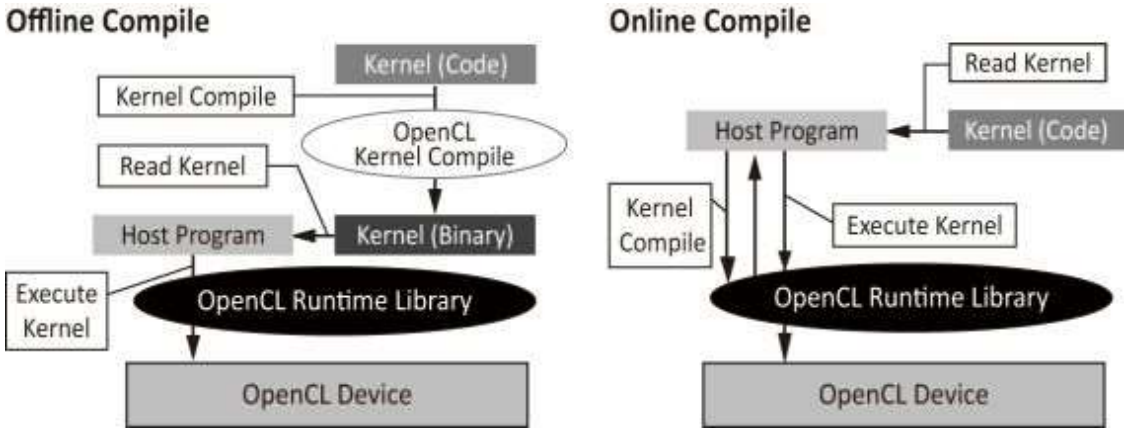
```
1.  #include <opencv2/ocl/ocl.hpp>

2.  int main(int argc, char** argv) {
3.   cv::Mat matIn = cv::imread("hoge.png"), matDisp;

4.   cv::ocl::oclMat oclIn(matIn), oclOut;
5.   cv::ocl::cvtColor(oclIn, oclOut, cv::COLOR_BGR2GRAY);
6.   oclOut.download(matDisp);

7.   return 0;
8.  }
```

SIProp

About OpenCL for OpenCV 2/2

SIProp

- Customized OpenCL for OpenCV
  - Header
    - opencv2/ocl/ocl.hpp
      - modules/core/src/opencl/runtime/generator/
  - Source Code
    - modules/core/src/opencl/*.cl
- OpenCL feature
  - NOT Binary Compatibility



Problem on Android

SIProp

- Required put *.CL files on same place of App
  - But Android App is APK file. Not single binary.
    - -> MUST build OpenCV.so w/your CL file

- “OpenCV with OpenCL for Android NDK”
  - How to Build OpenCV for Android System
    - https://github.com/noritsuna/OpenCVwithOpenCL4AndroidNDK

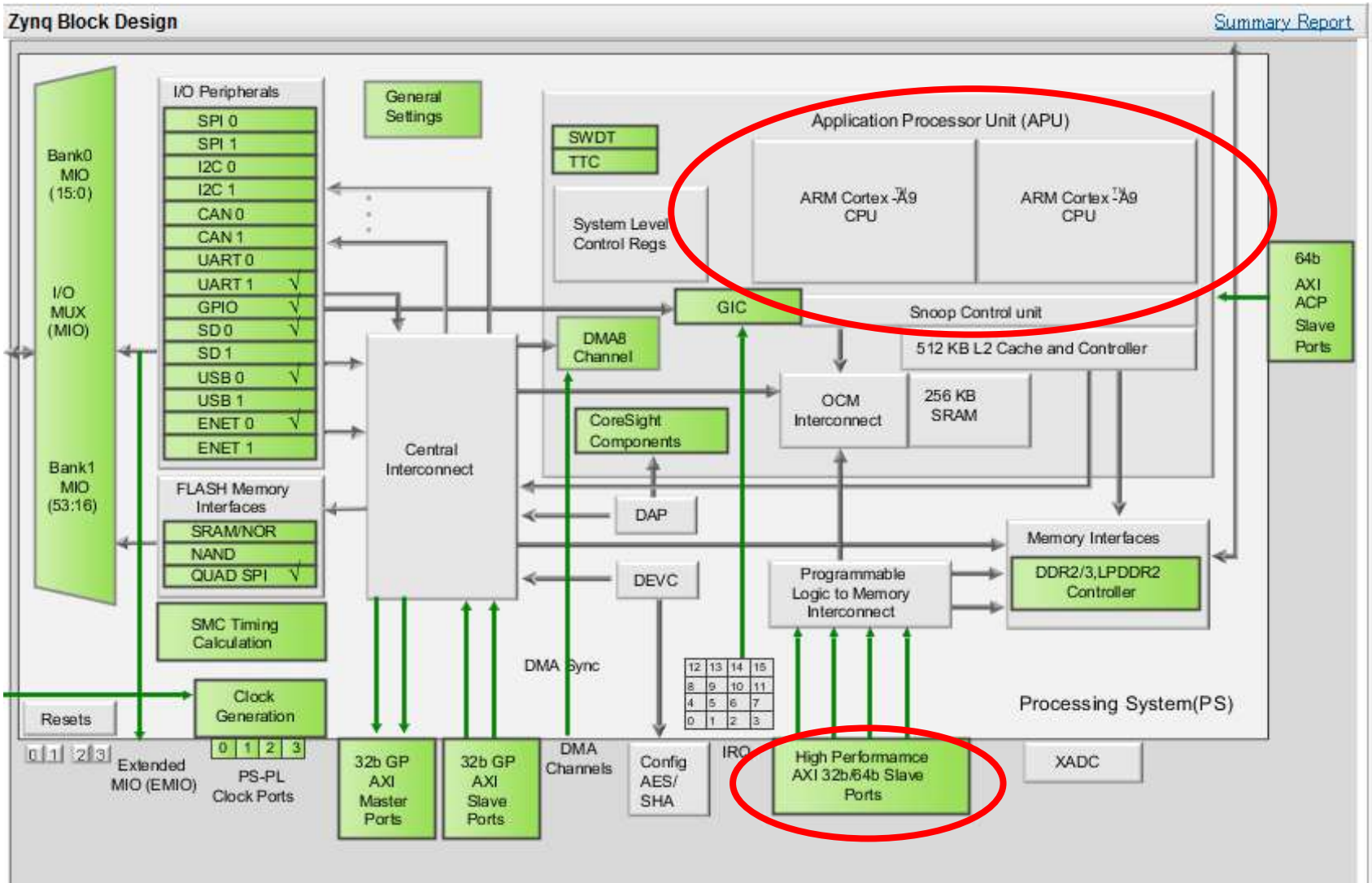
SIProp

- OpenCV for FPGA

Zynq-7000 Series

SIProp

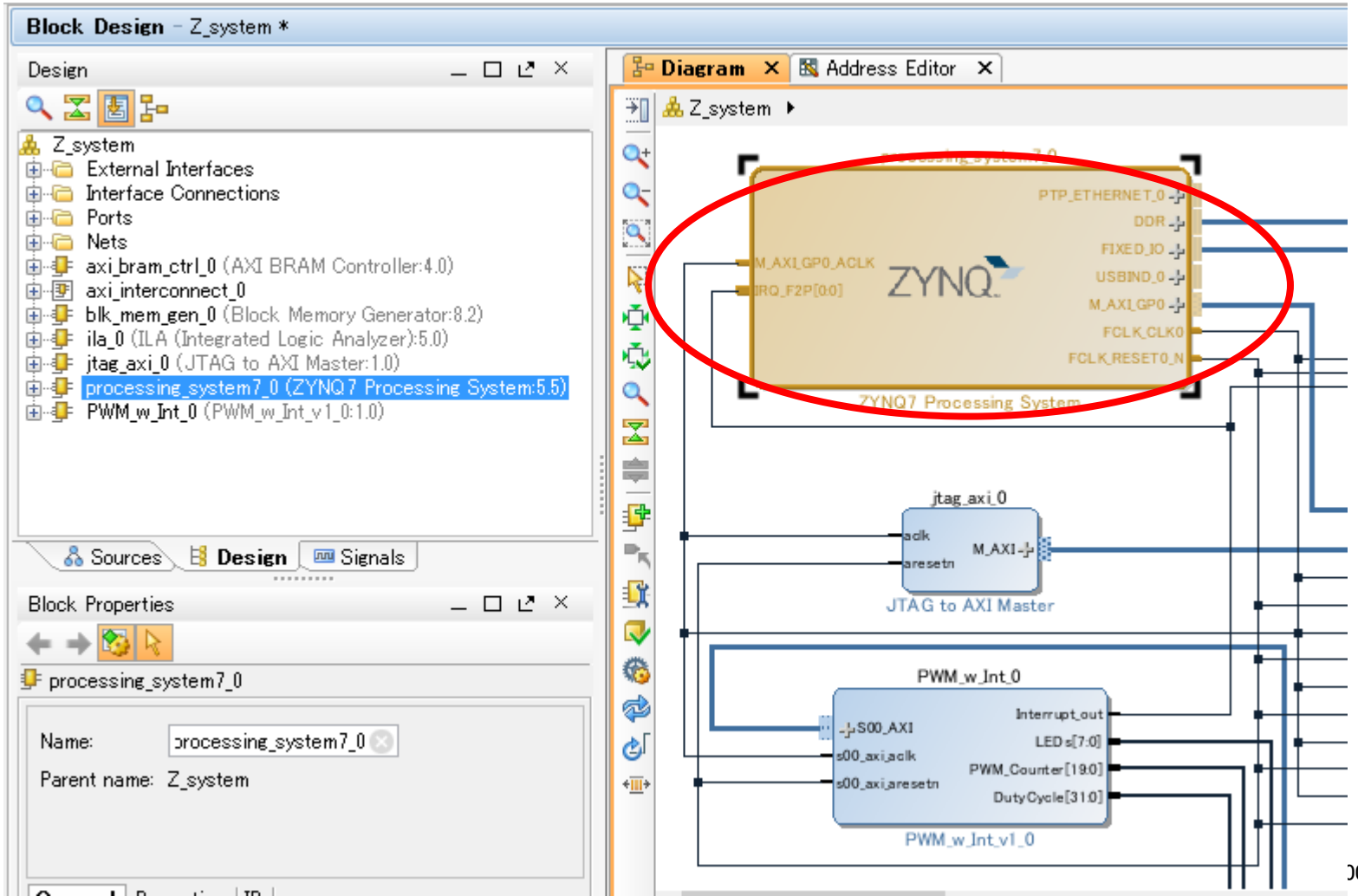
- Dual ARM Cortex-A9 + FPGA



Zynq + FPGA(IP Core)

SIProp

- Zynq + PWM IP Core



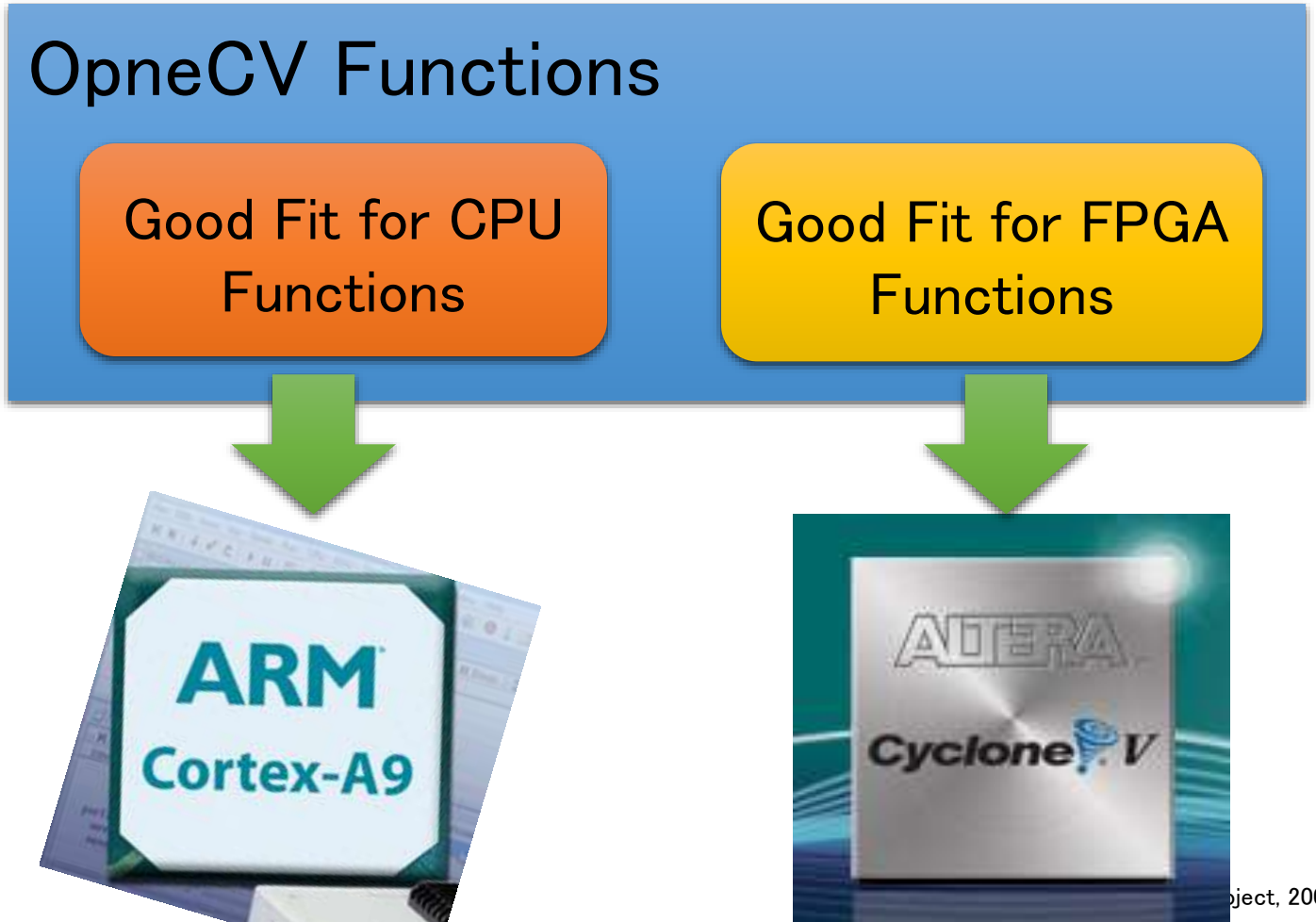
Zynq Development Board

- ZedBoard
  - USD495, Zynq-7020
- ZYBO
  - USD189, Zynq-7010
- Checkpoint
  - With Vivado(IDE) License?
    - http://www.digilentinc.com/

Advantage of ARM + FPGA

SIProp

- Full Functions of OpenCV
  - -> Required Super Large FPGA

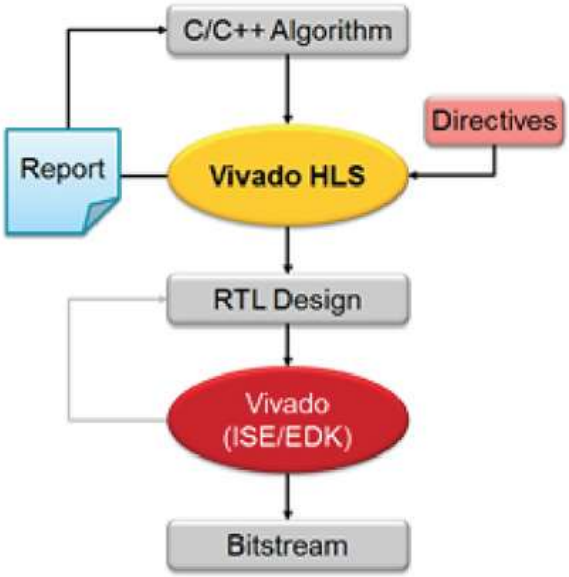


OpenCV Sample App for Zynq

SIProp

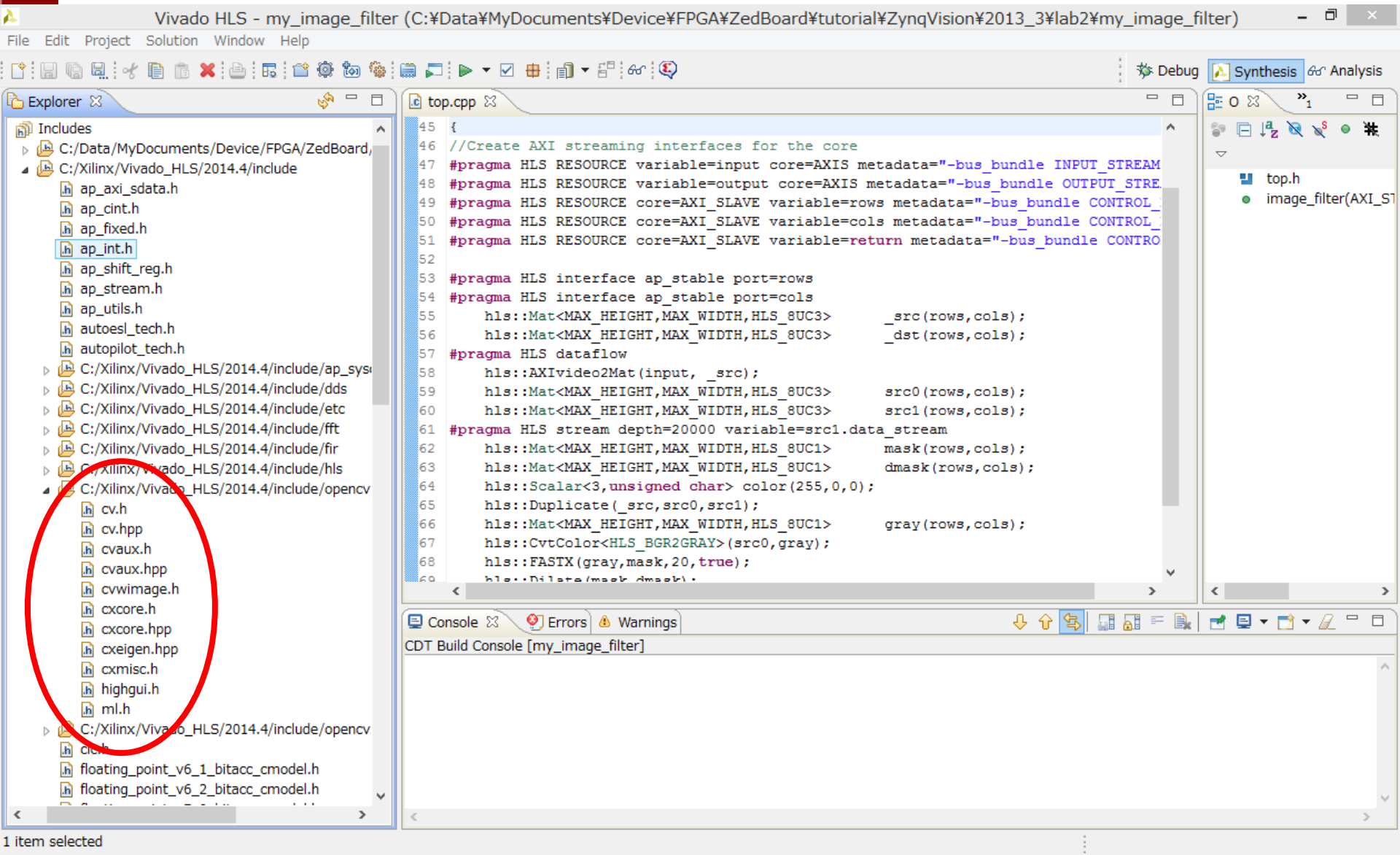
- "Accelerating OpenCV Applications with Zynq-7000 All Programmable SoC using Vivado HLS Video Libraries"
  - http://www.xilinx.com/support/documentation/application_notes/xapp1167.pdf

- Development Tools
  - Vivado HLS(High-Level Synthesizer)
    - Developing Environment for HLS
  - Vivado
    - IP Designer for Xilinx FPGA
  - ISE
    - Expired



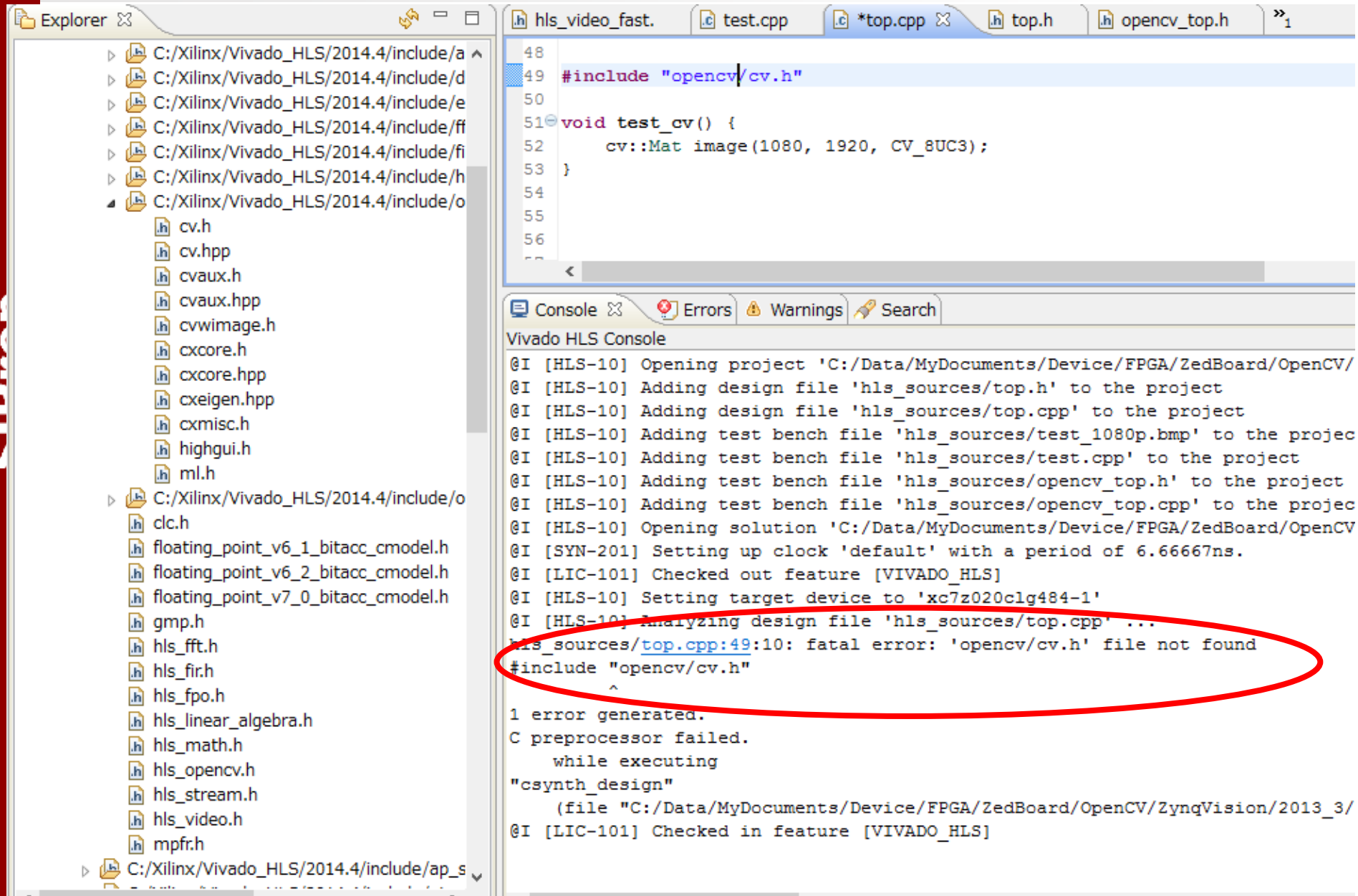
OpenCV is included in Vivado HSL

SIProp



Not Found…

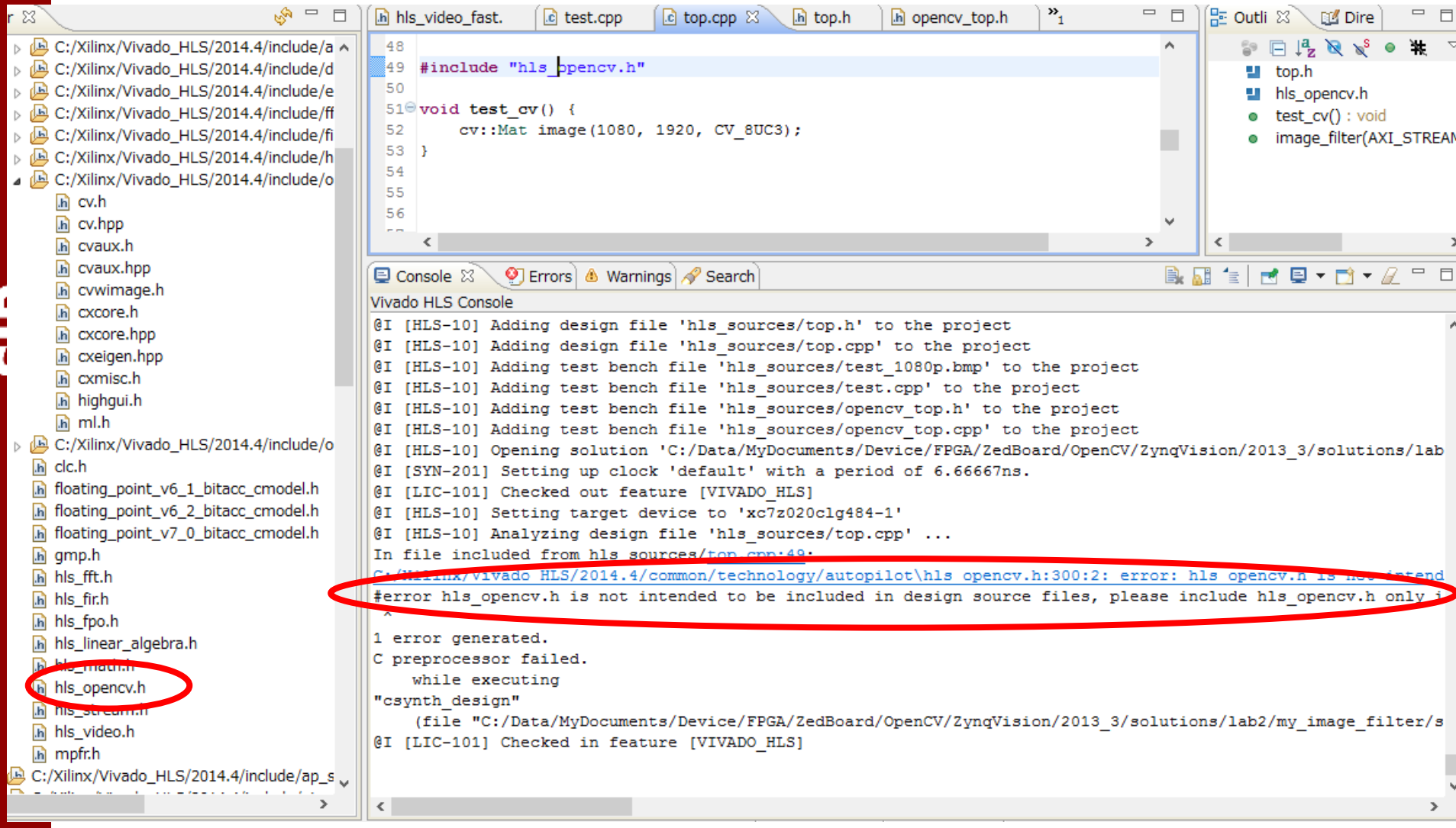
SIProp



Can't Use…

SIProp

- I guess "hls_opencv.h" is OpenCV for HLS.



Not Synthesizable…

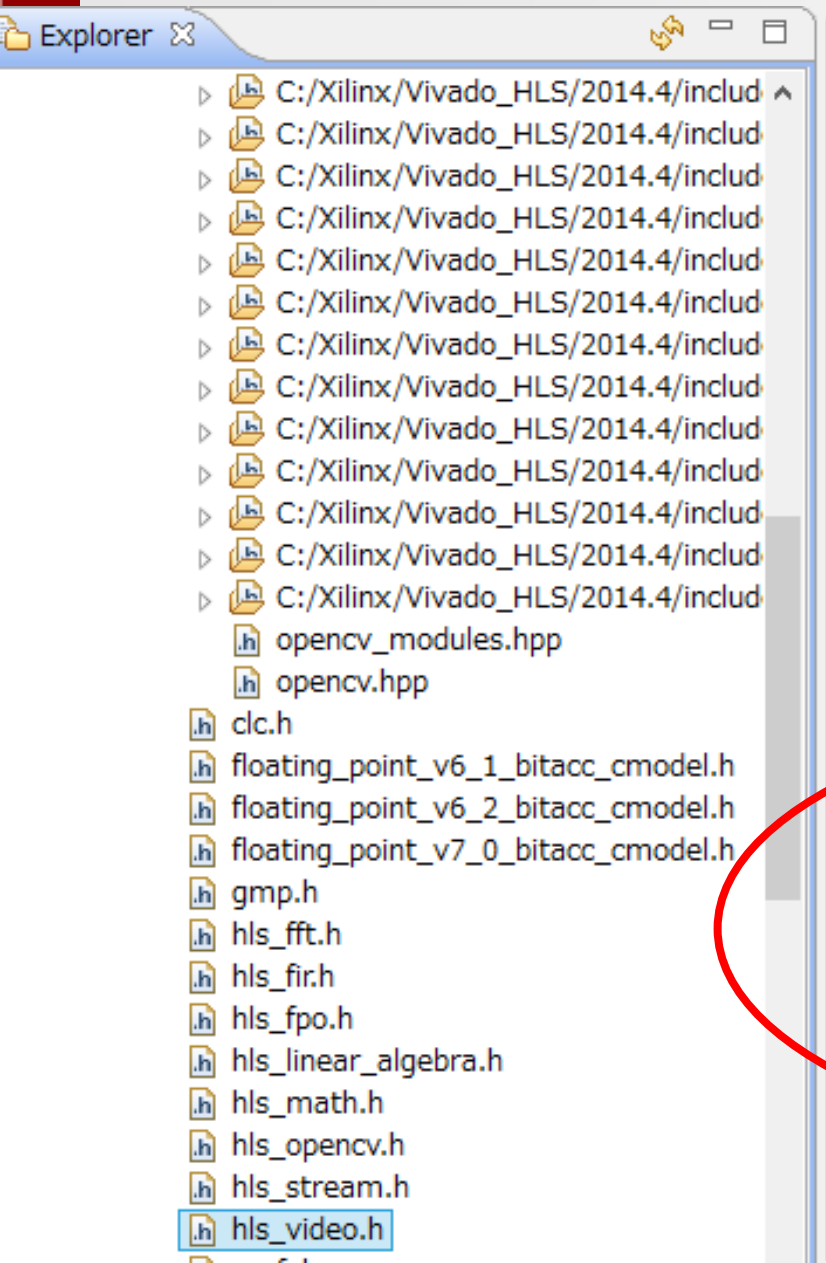
SIProp

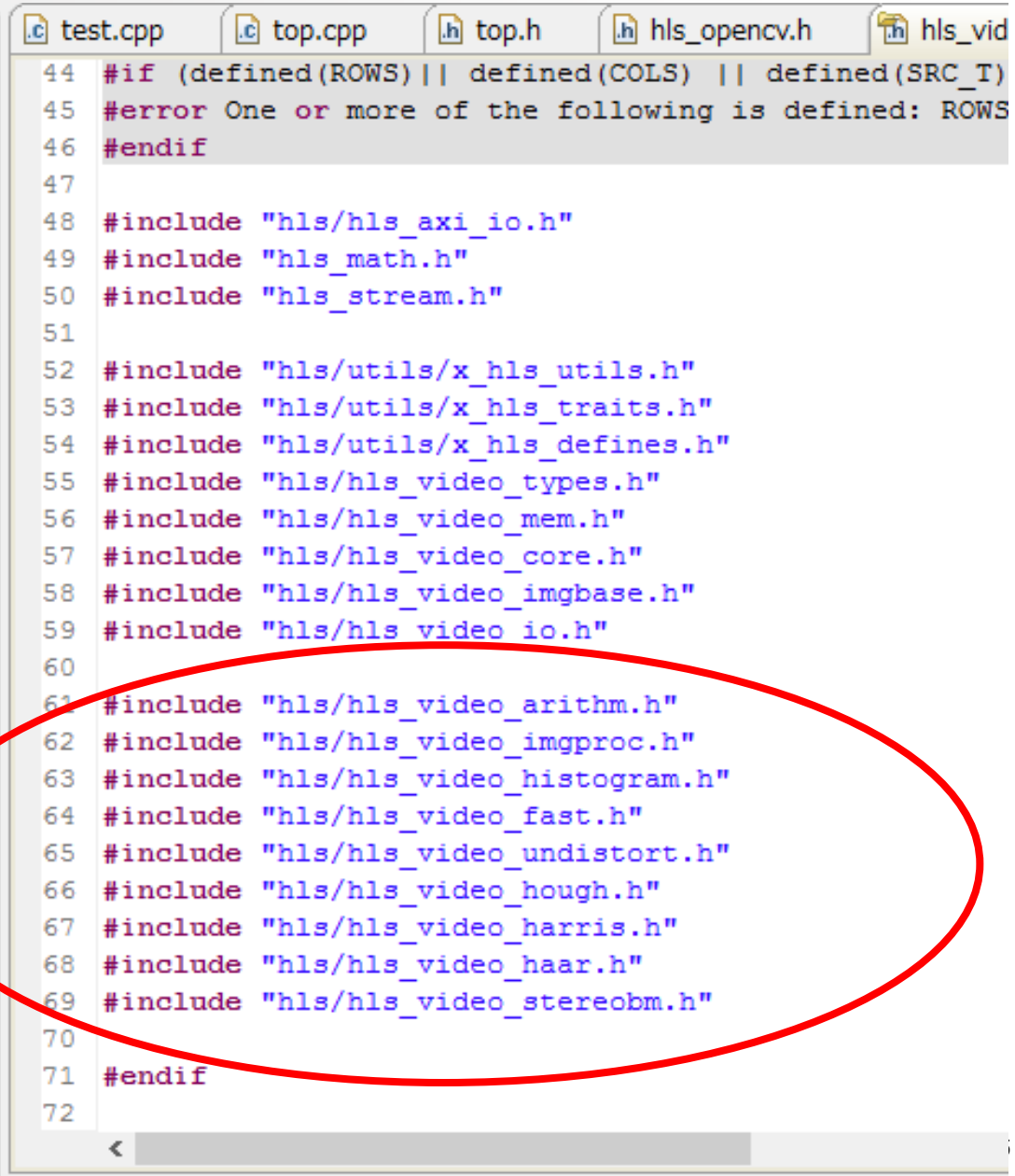
test.cpp | top.cpp | top.h | opencv_top.h | hls_opencv.h

```
 3  *      Author: Xilinx, Inc.
31
32 /*
33  * HLS OpenCV Image Translation File, Not Synthesizable.
34  */
35
36 #ifndef __HLS_VIDEO_OPENCV__
37 #define __HLS_VIDEO_OPENCV__
38
39 #ifndef __cplusplus
40 #error C++ is required to include this header file
41 #endif
42
43 //hls_opencv.h is only valid in testbench.
44 #ifndef __SYNTHESIS__
45
46 //OpenCV Header Files
47 #include "opencv2/opencv.hpp"
48
49 //HLS Video Header File
50 #include "hls_video.h"
51
52 /* From/To hls::Mat */
53
54 template<int ROWS, int COLS, int T>
```

“hls_video.h” has OpenCV for HLS

SIProp





OpenCV Sample App for Zynq

SIProp

- “Accelerating OpenCV Applications with Zynq-7000 All Programmable SoC using Vivado HLS Video Libraries”
  - http://www.xilinx.com/support/documentation/application_notes/xapp1167.pdf

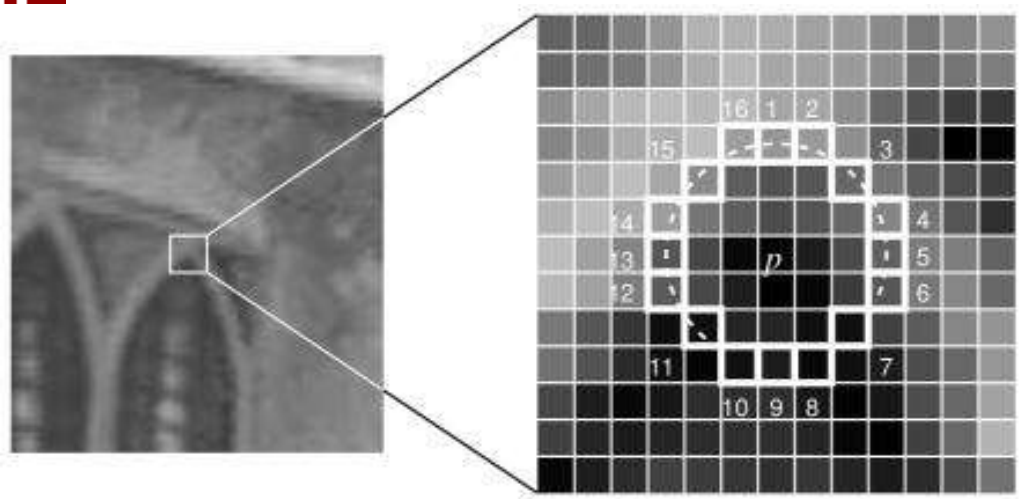
- Detail
  - Dilate Filter with FAST feature point for FullHD Video(1920x1080) Streaming

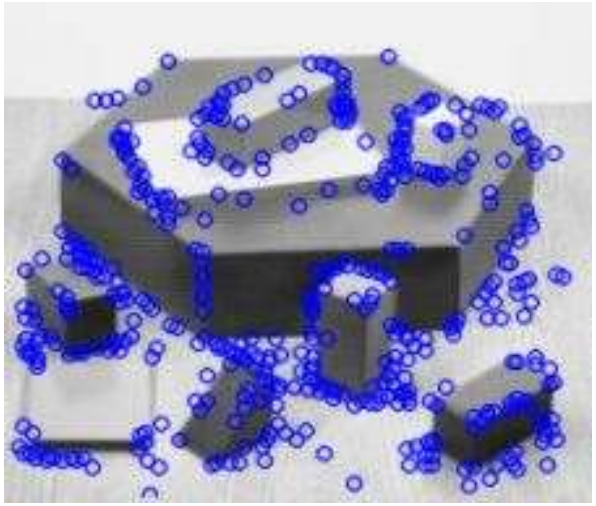
SIProp

```
1.  void image_filter(AXI_STREAM& input, AXI_STREAM& output, int rows, int cols)
2.  {
3.  //Create AXI streaming interfaces for the core
4.  #pragma HLS RESOURCE variable=input core=AXIS metadata="-bus_bundle
    INPUT_STREAM"
5.  #pragma HLS RESOURCE variable=output core=AXIS metadata="-bus_bundle
    OUTPUT_STREAM"
6.  #pragma HLS interface ap_stable port=rows
7.  #pragma HLS interface ap_stable port=cols
8.      hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     _src(rows,cols);
9.      hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     _dst(rows,cols);
10. #pragma HLS dataflow
11.     hls::AXIvideo2Mat(input, _src);
12.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     src0(rows,cols);
13.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     src1(rows,cols);
14. #pragma HLS stream depth=20000 variable=src1.data_stream
15.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     mask(rows,cols);
16.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     dmask(rows,cols);
17.     hls::Scalar<3,unsigned char> color(255,0,0);
18.     hls::Duplicate(_src,src0,src1);
19.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     gray(rows,cols);
20.     hls::CvtColor<HLS_BGR2GRAY>(src0,gray);
21.     hls::FASTX(gray,mask,20,true);
22.     hls::Dilate(mask,dmask);
23.     hls::PaintMask(src1,dmask,_dst,color);
24.     hls::Mat2AXIvideo(_dst, output);
25. }
```

FAST() function

- FAST (Features from Accelerated Segment Test) algorithm
  - One of the features detection algorithm





Dilate() function

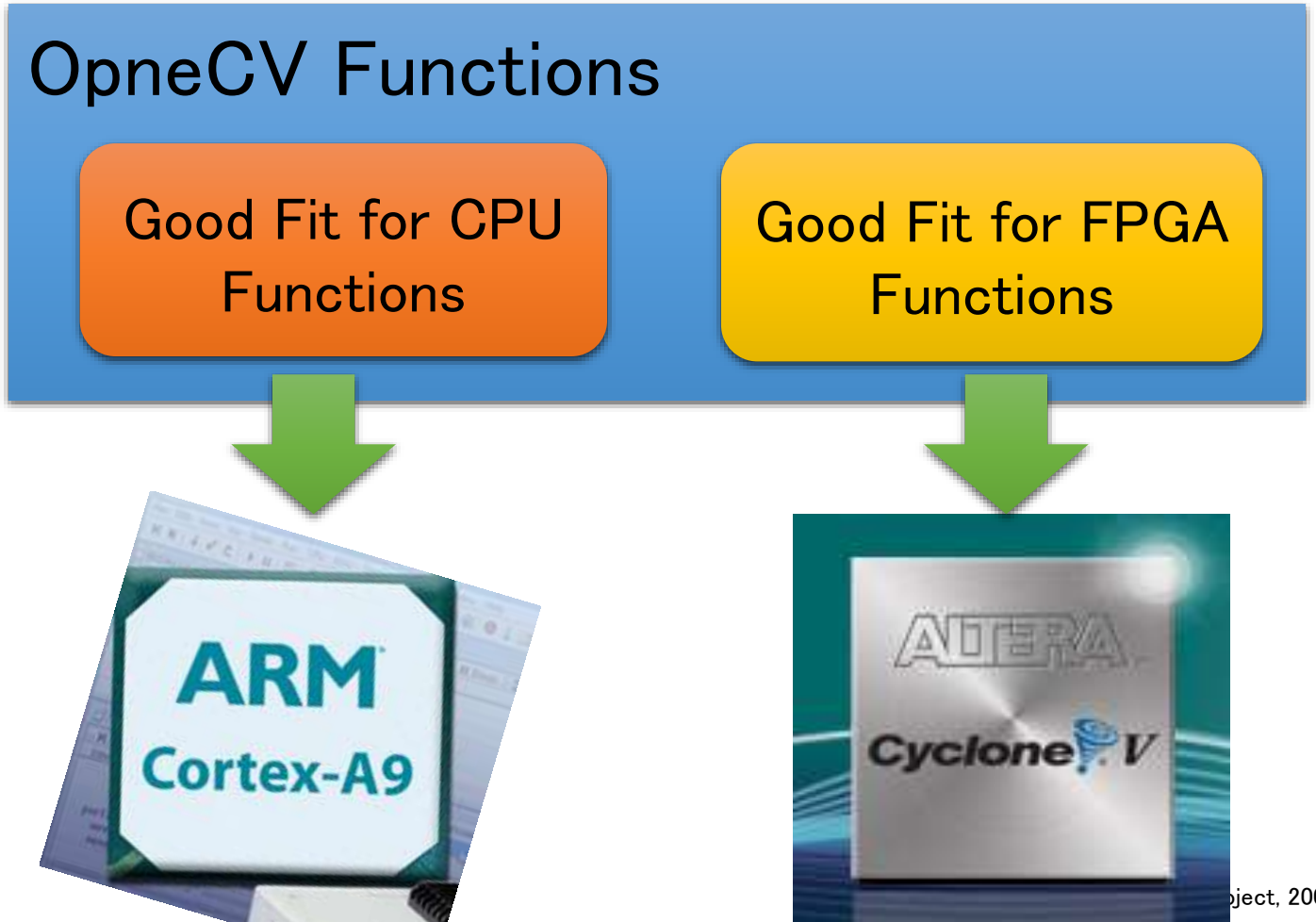
- Dilating pixels function
  - One of the filter function

```
In[1]:= Dilation[Dilation, {{1, 1, 1, 1}}]
Out[1]= Dilation
```

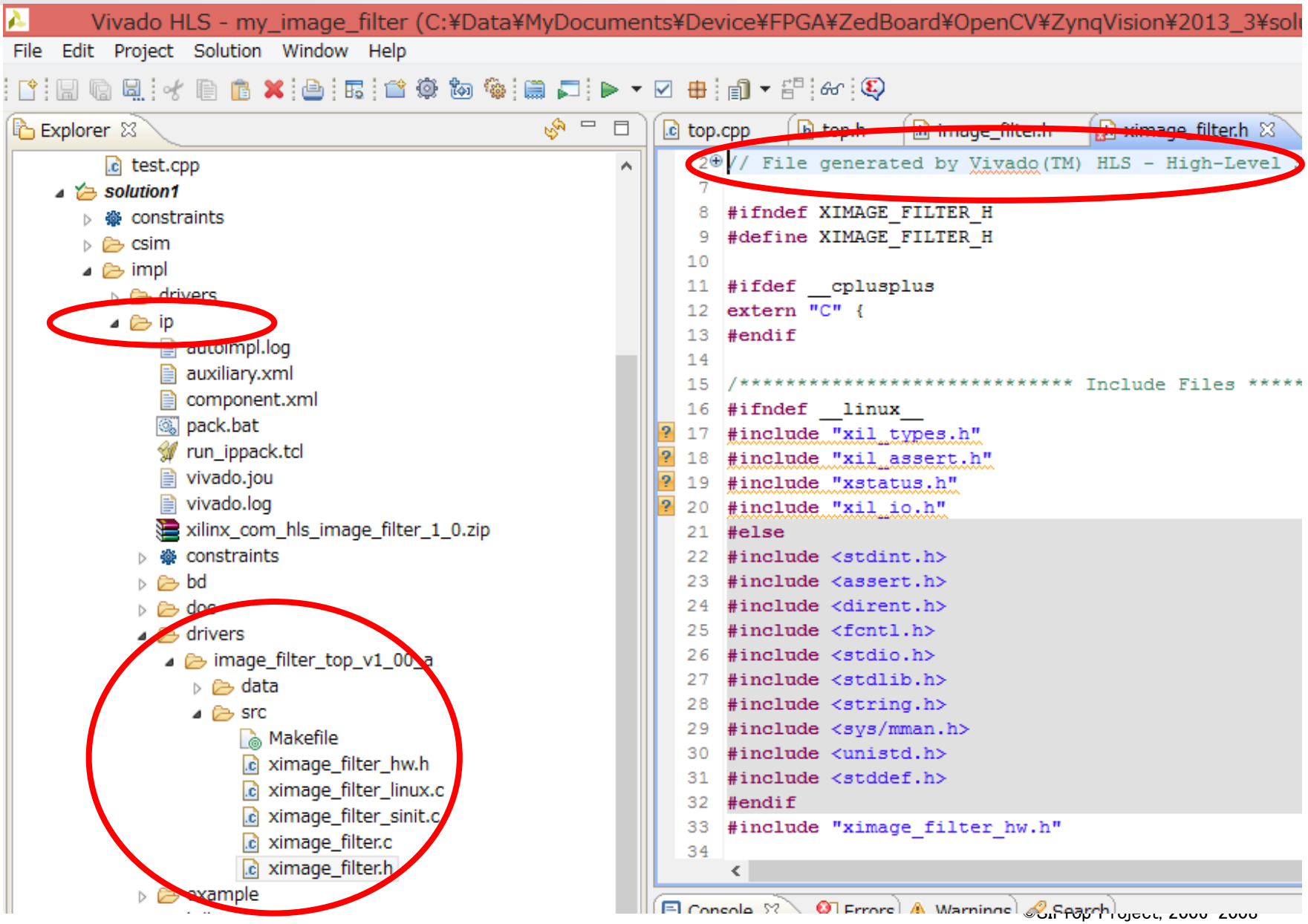
How to Implement ARM + FPGA?

SIProp

- Full Functions of OpenCV
  - –> Required Super Large FPGA



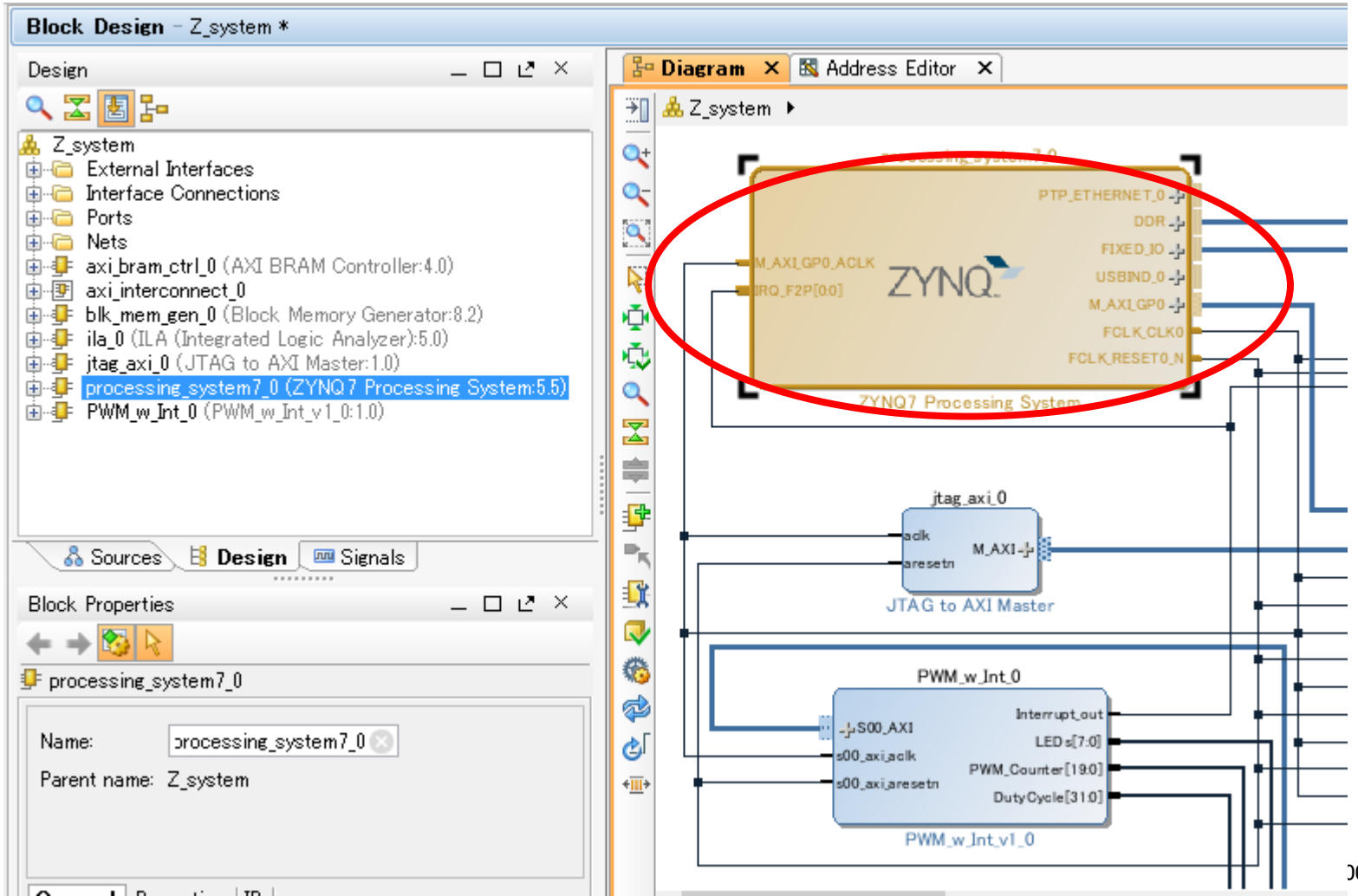
Generated “IP Core + Header”=“Driver”



Zynq + FPGA(IP Core)

SIProp

- Zynq + PWM IP Core on Vivado(≠Vivado HSL)



SIProp

```
1.  void image_filter(AXI_STREAM& input, AXI_STREAM& output, int rows, int cols)
2.  {
3.  //Create AXI streaming interfaces for the core
4.  #pragma HLS RESOURCE variable=input core=AXIS metadata="-bus_bundle
    INPUT_STREAM"
5.  #pragma HLS RESOURCE variable=output core=AXIS metadata="-bus_bundle
    OUTPUT_STREAM"
6.  #pragma HLS interface ap_stable port=rows
7.  #pragma HLS interface ap_stable port=cols
8.      hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     _src(rows,cols);
9.      hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     _dst(rows,cols);
10. #pragma HLS dataflow
11.     hls::AXIvideo2Mat(input, _src);
12.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     src0(rows,cols);
13.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     src1(rows,cols);
14. #pragma HLS stream depth=20000 variable=src1.data_stream
15.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     mask(rows,cols);
16.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     dmask(rows,cols);
17.     hls::Scalar<3,unsigned char> color(255,0,0);
18.     hls::Duplicate(_src,src0,src1);
19.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     gray(rows,cols);
20.     hls::CvtColor<HLS_BGR2GRAY>(src0,gray);
21.     hls::FASTX(gray,mask,20,true);
22.     hls::Dilate(mask,dmask);
23.     hls::PaintMask(src1,dmask,_dst,color);
24.     hls::Mat2AXIvideo(_dst, output);
25. }
```

How to USE “IP Core & Headers”

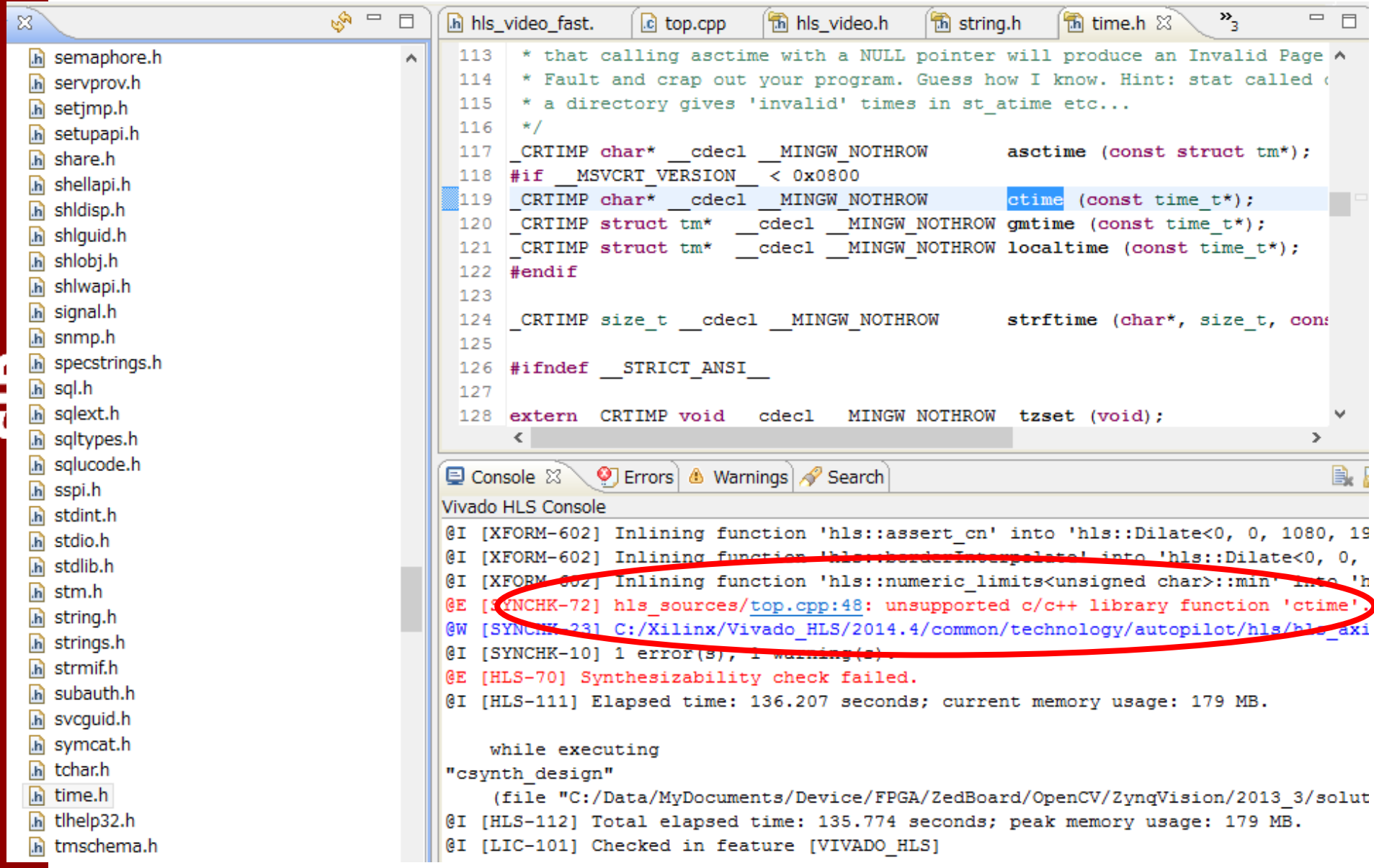
SIProp

- Build Binaries
  - FSBL(1st Boot Loader) = x-loader, u-boot
  - Linux Kernel for Your System
  - Ubuntu/Android for System (Option)
- How to Build Android 5.0 for ZedBoard
  - http://www.slideshare.net/noritsuna/zedroid-android-50-and-later-on-zedboard

```
1.  void image_filter(AXI_STREAM& input, AXI_STREAM& output, int rows, int cols)
2.  {
3.  //Create AXI streaming interfaces for the core
4.  #pragma HLS RESOURCE variable=input core=AXIS metadata="-bus_bundle
    INPUT_STREAM"
5.  #pragma HLS RESOURCE variable=output core=AXIS metadata="-bus_bundle
    OUTPUT_STREAM"
6.  #pragma HLS interface ap_stable port=rows
7.  #pragma HLS interface ap_stable port=cols
8.      hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     _src(rows,cols);
9.      hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     _dst(rows,cols);
10. #pragma HLS dataflow
11.     hls::AXIvideo2Mat(input, _src);
12.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     src0(rows,cols);
13.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC3>     src1(rows,cols);
14. #pragma HLS stream depth=20000 variable=src1.data_stream
15.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     mask(rows,cols);
16.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     dmask(rows,cols);
17.     hls::Scalar<3,unsigned char> color(255,0,0);
18.     hls::Duplicate(_src,src0,src1);
19.     hls::Mat<MAX_HEIGHT,MAX_WIDTH,HLS_8UC1>     gray(rows,cols);
20.     hls::CvtColor<HLS_BGR2GRAY>(src0,gray);
21.     hls::FASTX(gray,mask,20,true);
22.     hls::Dilate(mask,dmask);
23.     hls::PaintMask(src1,dmask,_dst,color);
24.     hls::Mat2AXIvideo(_dst, output);
25. }
```

Can't Use All Standard C/C++ Libs

SIProp



SIProp

OpenCV acceleration battle:
OpenCL on Firefly-RK3288(MALI-T764)
vs.
FPGA on ZedBoard(Zynq-7020)

Noritsuna Imamura

noritusna@siprop.org

Which way is faster?

- A. OpenCV for FPGA >>>> OpenCV for OpenCL
  - According to Xilinx “1000 times over faster”.
  - Clock Speed
    - FPGA = 150MHz, ARM-T764 = 650MHz
- Direct connect Peripherals to FPGA.
  - GPGPU must bypass CPU/Memory bus.