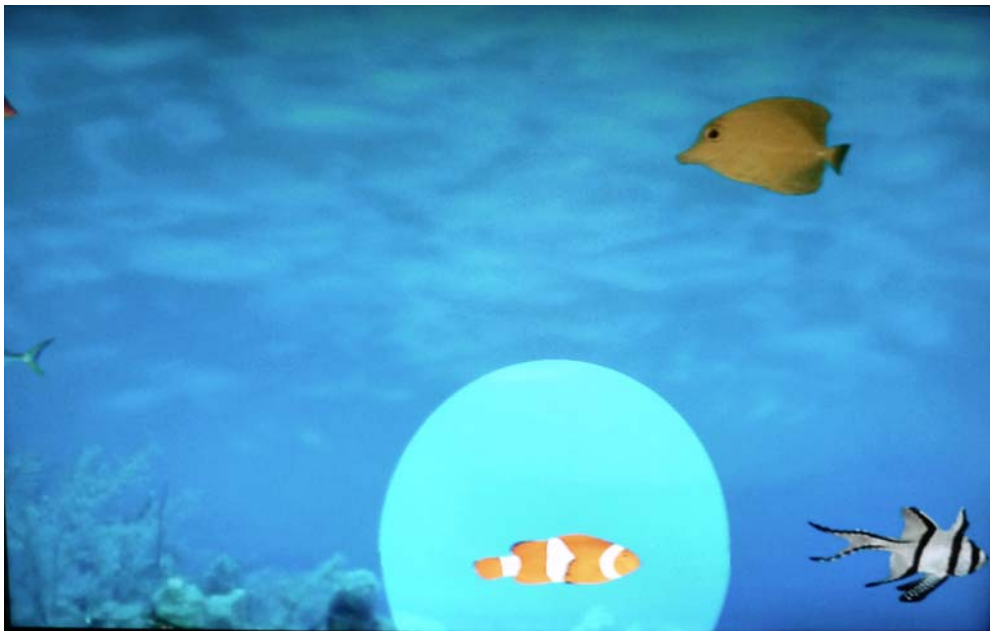
Implementing an HD-Video Control Plane Embedded Processing System For Object Detection and Highlighting



This Page Intentionally Left Blank

Lab Overview

This workshop is an introduction to using the Xilinx® MicroBlaze™ embedded processor for video control plane applications. The Xilinx Video Starter Kit, Spartan™ 3A DSP Edition is used in the design implementation. In this hands-on workshop you will build a complete streaming video application that includes DVI interfaces, a run-time configurable video filter, a real-time object detection and tracking filter, and a MicroBlaze™ embedded processor for embedded control of the video subsystem.

First, you will build and deploy a streaming DVI video pass-through example, using the Xilinx Embedded Development Kit tools and a Xilinx-provided certified reference project. Upon completion of this first project you will be able to observe HD video data passing through the FPGA. You will learn during this first project how to use the embedded MicroBlaze processor as an embedded video controller.

Next, you will augment the pass-through test application project by adding a run-time configurable 2D FIR filter that has been created using the Xilinx System Generator™ tools. During this part of the workshop you will gain hands-on experience with how to add and connect IP using Xilinx Platform Studio™.

Finally, you will include a second video processing component that has been developed using C-to-FPGA tools. The resulting video project will demonstrate a moderately complex object recognition and tracking algorithm that allows specific objects to be identified and highlighted in the output video. You will observe the results of your efforts after building and downloading the completed project.

Figure 1 illustrates the completed system, with DVI input and output, pipelined video filters and the MicroBlaze™ processor.

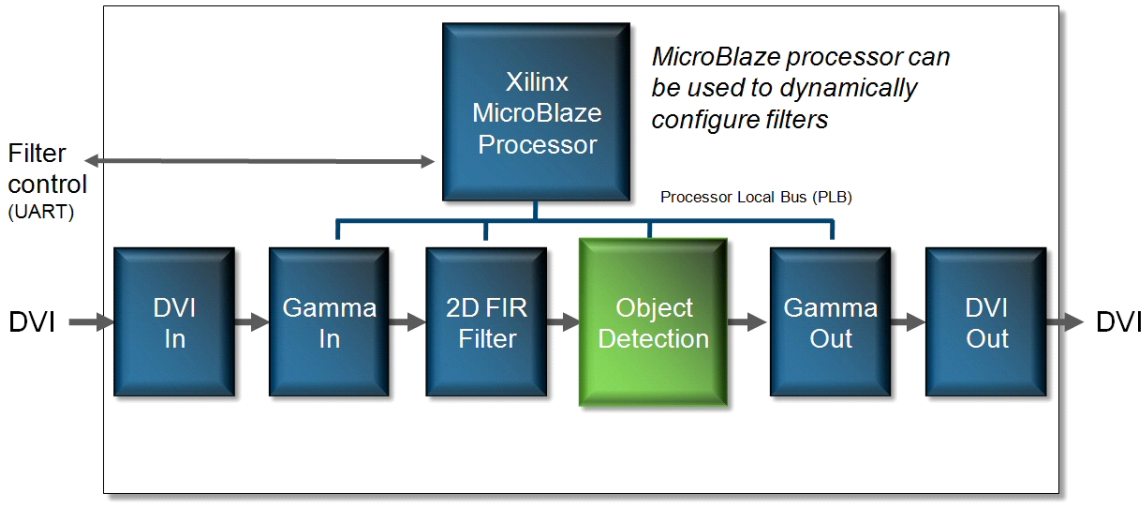


Figure 1 - Streaming Video Block Diagram

The completed system will process video data at 720p resolution (1280 by 720 pixels). The filters will analyze the video stream and look for a distinct pattern, for example the stripes of a fish, then produce a moving spotlight effect around the detected target. The components of this streaming video example include:

- MicroBlaze soft processor and related peripherals including a UART for serial communications
- DVI input and output interfaces
- Gamma-in and Gamma-out blocks for color format conversion
- 2D FIR filter block previously developed using Xilinx System Generator™
- Detection and highlighting filter previously generated by a C-to-FPGA compiler

Objectives

After completing this workshop, you will be able to:

- Build and download a MicroBlaze processor based-system using the Xilinx Platform Studio tools.
- Create and modify software and hardware components within Platform Studio.
- Use Xilinx-provided certified reference designs and the Xilinx Spartan 3A Video Starter Kit.
- Implement a complete streaming video subsystem on a Xilinx FPGA.
- Add Xilinx-provided and custom filters to the video processing pipeline.
- Understand how to combine Xilinx System Generator, Xilinx Platform Studio and third party tools for increased productivity.

Downloading the Design Files

The design files used in this workshop can be downloaded from the Xilinx SPW website.

Procedure

This lab consists of the following steps:

- Start with a pre-built video pass-through reference design and verify its operation.
- Use Xilinx Platform Studio to import and integrate a 2D FIR filtering component previously generated using Xilinx System Generator™.
- Connect the 2D FIR filter clock and reset, and connect the PLB bus interface.
- Download the 2D FIR project and observe the video pass-through behavior.
- Use the MicroBlaze controller, via a serial cable connection, to modify the FIR filter coefficients during run-time.
- Walk through the C-code for the Object Detection and Highlighting filter.
- Import the pre-built objection and detection hardware filter into Xilinx Platform Studio, connecting it into the existing reference design using the DVI and PLB bus interfaces.
- Build the project and download the completed design.
- Use the serial cable connection to configure the detection and highlighting filter.

Requirements

Hardware Requirements

- Xilinx Video Starter Kit (VSK), Spartan 3A Edition
- DVI-capable video monitor capable of displaying 720p (1280 X 720) resolution
- DVI Video source capable of outputting 720p resolution
    - Design has been tested with DViCO TViX HD M-6500A video player http://www.dvico.com/
- Workstation running Windows XP or Linux
    - This module describes the design running on a Windows XP system
- Xilinx Platform Cable USB for downloading a design
- Serial cable provided with the VSK

Software Requirements

- Xilinx ISE and EDK 11.1i software
- ZIP file: MicroBlaze_VSK_FishFinder.zip containing the design files
- HyperTerminal or similar terminal program running at 9600 baud

Initial Setup

- Unzip the file MicroBlaze_VSK_FishFinder.zip to C:
- This will place the design files in several directories under:
  c:\xilinx_design\S3A_VSK_Labs\FishFinder\
- The aquarium video to be played will be placed at:
  c:\xilinx_design\S3A_VSK_Labs\FishFinder\Aqua1280x720x4.wmv
- The workshop instructions will be placed at:
  c:\xilinx_design\S3A_VSK_Labs\FishFinder\
  MicroBlaze_VSK_FishFinder_Lab_11.1.pdf
- Connect the Xilinx USB download cable to the workstation and VSK board
- Connect the serial cable between the workstation and the VSK board
- Connect a DVI cable between the VSK DVI output port (located on the main board) and the monitor
- Connect a DVI cable between the VSK DVI input port (located on the daughter card) and the DVI video source
- Copy the video source to the player, start the video source playing the aquarium video, and set for continuous repeat

Getting Started

Figure 2 shows the Xilinx Video Starter Kit, Spartan 3A Edition used for this workshop.



Figure 2 – Xilinx Video Starter Kit

We will be using the DVI input and output interfaces for this example to filter and analyze a 720p format (1280 by 720 pixels) video signal. Figure 1 shows a detailed system diagram of the final video filtering design to be implemented. Note that the complete system includes an Embedded MicroBlaze processor that is connected to the Gamma In, Gamma Out, 2D FIR and Object Detection cores using on-chip bus interfaces.

You will create this example in three steps. First, you will load and test a pre-built video pass-through example provided with the Video Starter Kit. This pre-built example will allow you to verify that your hardware is properly set up, and that you have a good starting point from which to create more complex filtering applications.

After successfully building and testing the pass-through reference design, you will then add a new hardware FIR filter (step 2) and then a filter for objection detection (step 3) into the video processing pipeline. You will also modify the software portion of the application (running on the MicroBlaze processor) to communicate with the new filter, allowing run-time configuration.

Implementing the Design via Xilinx Platform Studio

Part 1 – Verify the Video Hardware Setup



You will begin by running a quick test of the video hardware, using a pre-built pass-through reference example provided on the Flash memory card on the VSK board. This pass-through example includes the same video processing components shown in Figure 1, with the exception of the object detection component.



1. Start by verifying the video stream without the VSK board attached:
   - Ensure that the DVI cable is directly connected from the video source (the TViX unit) to the monitor.
   - If necessary, power on the video source and select the Aqua1280x720x4.wmv video file.
   - Press the Repeat button on the video remote control to cause the video file to loop.
   - Verify that video data is appearing on the monitor.
2. Test the VSK board for correct setup:
   - Leave the video source powered up and the video playing.
   - Disconnect the DVI source cable from the monitor, and reconnect the DVI source cable to the input DVI connector on the VSK board. The input DVI cable is the one on the small daughter card.
   - Connect the DVI output cable from the VSK board output DVI connector, to the monitor.
   - Assure that the VSK board is configured to its default settings and the SystemAce compact flash card is installed.
   - Turn on the VSK board power. On the VSK board you will see a menu allowing you to select from a number of demonstrations.
   - Select the first demonstration (DVI Pass-through Demo) by pressing the push-button located in the **center** of a cluster of five buttons, near the small LCD display on the VSK board.
   - Verify that the video is being properly displayed on the monitor.

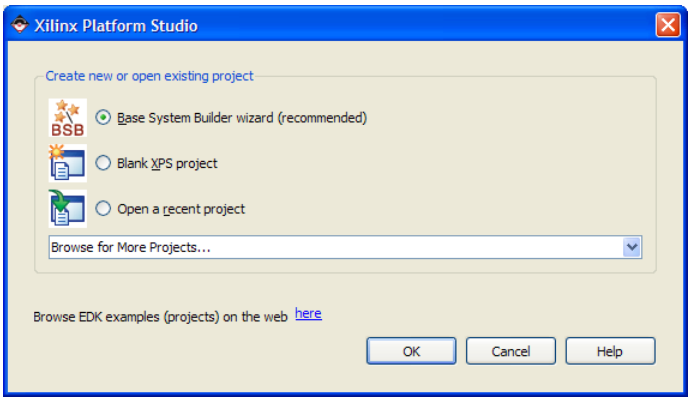
Part 2 – Build the Pass-Through Example Using Platform Studio

In this step, you will learn how the video pass-through example was created and assembled using the Platform Studio tools, in combination with System Generator. To do this, you will load a partially-completed reference example. You will then add and connect an existing component (the 2D FIR filter) that was previously designed and created using System Generator.

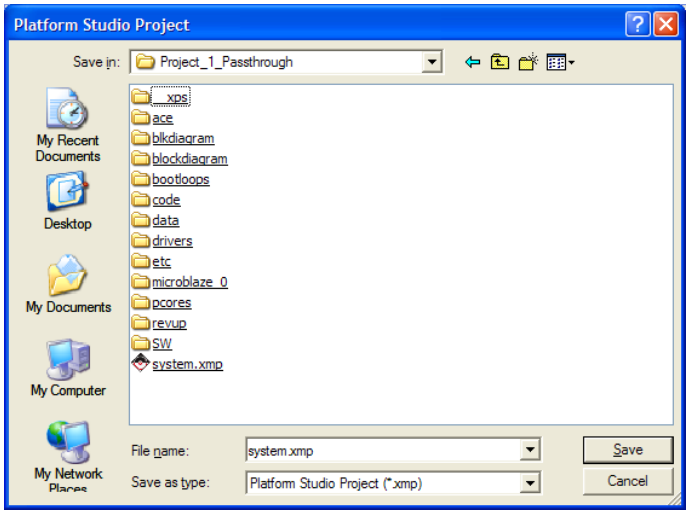
The complete system you will build will include:

- MicroBlaze processor core
- Processor Local Bus (PLB)
- Internal FPGA Block RAM
- GPIO interface to accept button inputs
- UART as input/output device
- Peripheral cores (pcores) generated using System Generator, including:
  - DVI input and output interfaces
  - Configurable Gamma In and Gamma Out filters
  - Configurable 2D FIR filter

1. Open Xilinx Platform Studio:
   a. Double-click on the Xilinx Platform Studio icon on the desktop
   b. **Select** the **Open a recent project** option.



   c. **Select Browse for More Projects** as shown.
   d. **Click OK**.
   e. **Navigate** to the directory: **c:\xilinx_design\S3A_VSK_Labs\FishFinder\Project_1_Passthrough**
   f. **Select System.xmp**.



g. **Click Open**

- After a few seconds, the project will appear.

h. Towards the bottom-center of the screen **click** on the **System Assembly View** tab to obtain the view as shown in Figure 3.

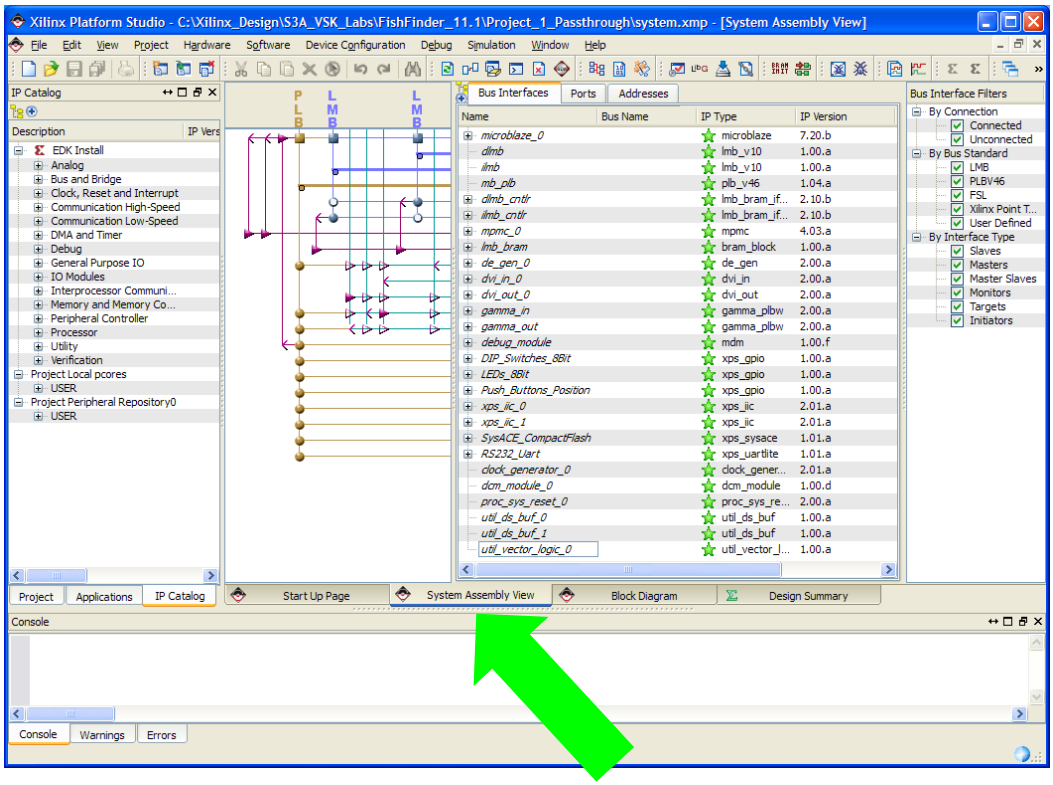


Figure 3 – Platform Studio with Video Pass-Through Project

The Platform Studio project file that you have just opened is similar to the pre-loaded reference example provided on the Flash card. In this version, however, we have intentionally left out one of the filter components, the 2D FIR filter. Before adding the FIR filter core, let's take a few moments to review the Platform Studio Application.

In the Platform Studio graphical interface, you can see three windows, each of which has three tabs. These three windows are:

- The **Project Information Area** on the left side: including Project, Application and IP Catalog tabs. This window provides a summary of your project, and is where you can easily add or delete various IP components. This is also where you can manage the software portion of your application including source code that will be compiled to the embedded MicroBlaze processor.
- The **System Assembly View** on the right side: which provides a visual indication of how the various IP components are connected (via bus interfaces or wires) in the system.
- A **Console Window** at the bottom: where various messages, including warnings and errors, are displayed during processing.

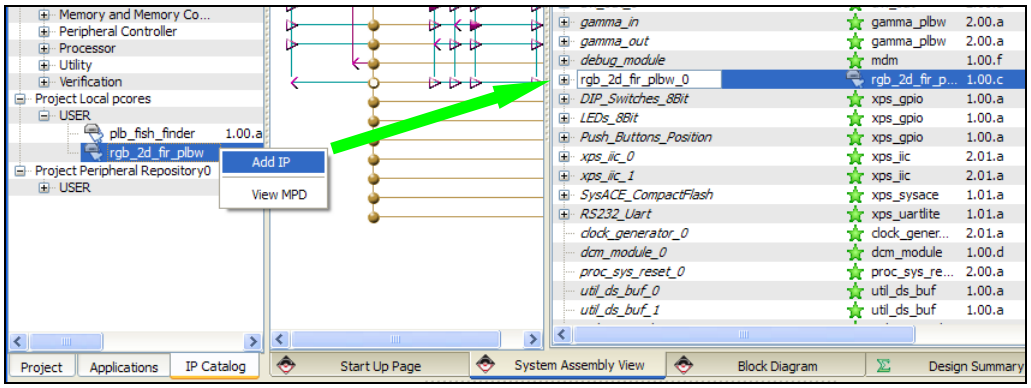
2. Use the following steps to add the 2D FIR filter to the project and attach it to the video pipeline and PLB bus interfaces:

   a. In the Project Information Area window, **click** on the **IP Catalog** tab to view available IP cores that can be added to your project.

   b. Find the category listed as "**Project Local pcores**", and **click** once on the small ⊞ icon next to the **USER** entry. You will find a peripheral core (pcore) named **rgb_2d_fir_plbw**.

      Note: This pcore was previously generated using the System Generator tools, and then copied into the project directory. We will use the same method later in the workshop to add a pcore generated using a C-to-HDL tool.

   c. **Right-click** on the **rgb_2d_fir_plbw** pcore and **select Add IP**. You will see the pcore appear in the System Assembly View as shown below:



The pcore must now be attached to the processor local bus, and must also be connected to the **Gamma In** and **Gamma Out** pcores via DVI bus interfaces.

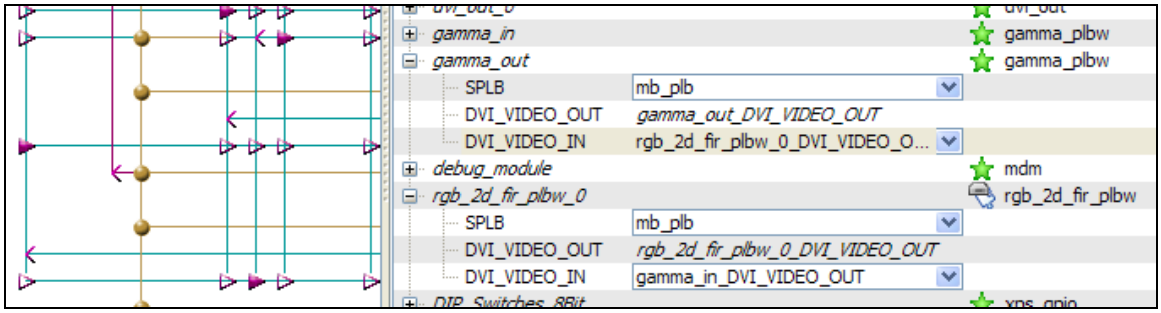
d. Expand the component **rgb_2d_fir_plbw_0** by **clicking** on the small ⊞ icon to view the available connections.

e. Connect the **rgb_2d_fir_plbw_0** component to the MicroBlaze PLB bus by **clicking** once on the small open circle to the left of the SPLB entry. This will establish the **rgb_2d_fir_plbw_0** as a slave on the PLB.

f. Connect the DVI input to the 2D FIR filter by **clicking** on the **DVI_VIDEO_IN** drop-down list, and **select gamma_in_DVI_VIDEO_OUT.**

g. A bus was automatically generated connected to the DVI output of the FIR filter. Connect the DVI output of the 2D FIR filter to the Gamma Out pcore by **clicking** on the small ⊞ icon next to the **gamma_out** pcore entry, then **click** on the **DVI_VIDEO_IN** drop-down list and **select rgb_2d_plw_0_DVI_VIDEO_OUT**.

h. The **rgb_2d_fir_plbw_0** component has now been attached to the processor local bus (PLB) and to the video pipeline via the DVI interfaces. The connections should appear as follows:



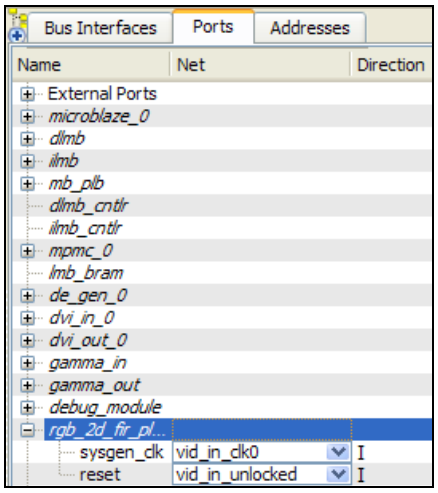
3. Connect the rgb_2d_fir clock and reset signals as follows:

    i. **Select** the **Ports** tab of the **System Assembly View**.

    j. **Click** the small ⊞ icon at **rgb_2d_fir_plbw_0** to expand the entry.

    k. For the **reset** connection; use the drop-down selection list to **select** the **vid_in_unlocked** signal

    l. For the **sysgen_clock** connection; use the drop-down selection list to **select vid_in_clk0**.

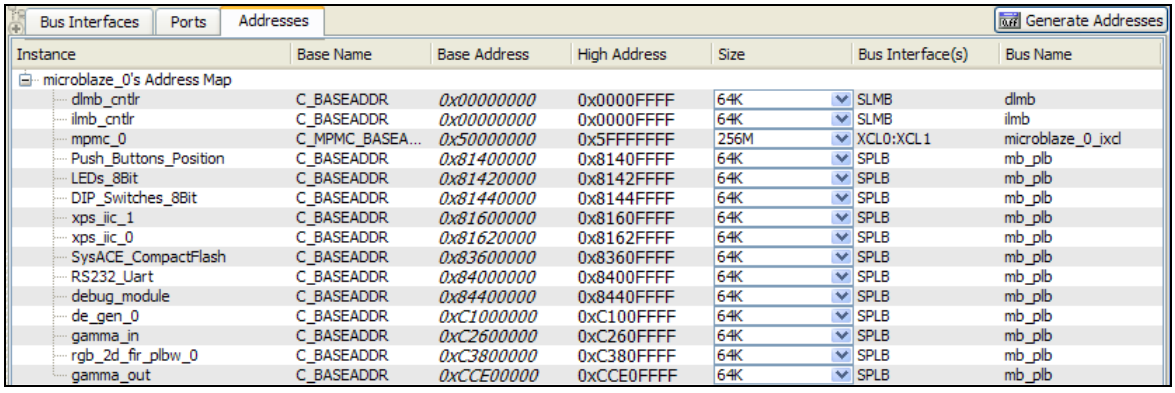


4. Create the base address for the added pcore as follows:

    a. **Click** on the **Addresses** tab of the System Assembly View.

    b. In the upper right corner, **Click** the **Generate Addresses button** to automatically assign a Base Address and Size to the rgb_2d_fir_plbw_0 pcore.

    c. The IP addresses will appear similar to below:



Part 3 – Build and Test the Pass-Through Example



In this section you will test the video pass-through example.

1. To save time and skip the build step, go directly to a pre-built design as follows:
   a. Select on the Menu Bar:
      File ⇒ Open Project
   b. Browse to and select the following file:
      c:\xilinx_design\S3A_VSK_Labs\FishFinder
      Project_1_Passthrough_Completed\system.xmp
      Hint: The desired directory should be up one level from the current directory.
   c. Click Open
   d. If a pop-up window prompting if it is OK to close the existing project appears, **Click YES**
2. Build the MicroBlaze software application as follows:
   a. **Select** on the **Menu Bar**:
      Software ⇒ Build All User Applications
   b. The build will take about 1 minute.
3. Start a terminal session:
   a. Confirm the serial cable is connected between the workstation and the VSK board
   b. Run the HyperTerminal program
   c. Select Serial Interface, COM1
   d. Specify 9600 baud, 8-bits, no parity, no flow control
4. Download and verify the application.
   a. Confirm that the Platform Cable is correctly connected to the workstation and VSK board
   b. To configure the FPGA design**, Select** on the **Menu Bar**:
      Device Configuration ⇒ Download Bitstream
   c. To download and run the software design, Select on the Menu BAR:
      Debug ⇒ Launch XMD

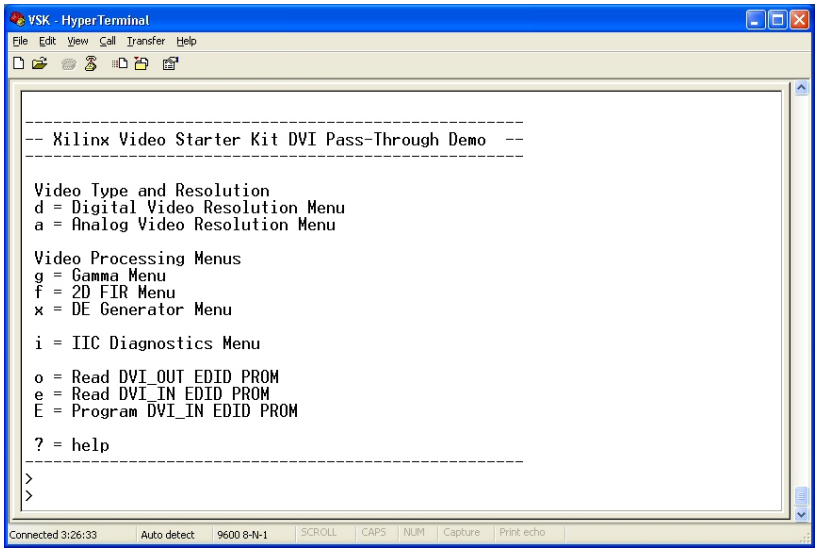
Note: We have modified the XMD command file to automatically download and run the software.

d. Close the XMD command window by **typing** "**exit"** at the prompt.

e. At this point you should see the video appearing on the output display monitor.

f. Use the HyperTerminal program to interact with the video filters.

- **Press** the **Enter** key to see a menu as shown below
- **Press** the **'f'** key to display and specify various 2D FIR filter options
- **Type a number from 1 to 9 and see the filter effect**
    - **Number 2 presents an interesting edge filter effect**



Part 4 – Add the Fish Finder Core to the Project

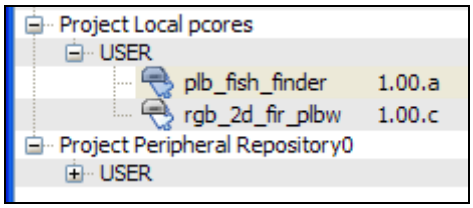
You will now repeat the steps in Part 3 to add an additional filter for object detection and highlighting. The Fish Finder filter was developed using C-to-FPGA tools.

1. Now add the object recognition and high-lighting filter as follows:

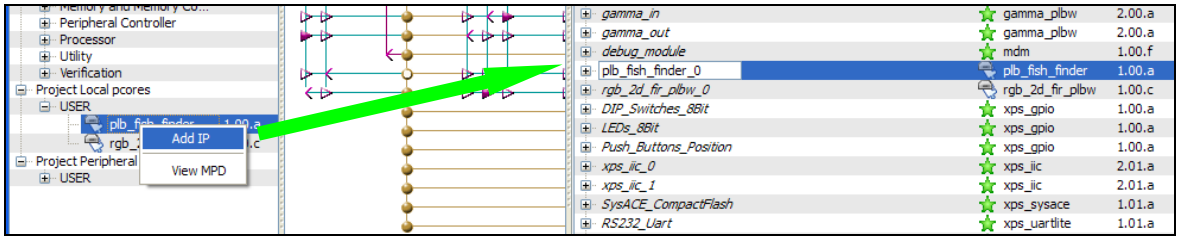
   a. Towards the center of the screen **click** on the **System Assembly View** tab to open the System Assembly View.

   b. In the Project Information Area window, click on the IP Catalog tab to view available IP cores that can be added to your project.

   c. Find the category listed as "**Project Local pcores**", and **click** on the small ⊞ icon next to the **USER** entry. You will find a peripheral core (pcore) named **plb_fish_finder**:



   d. **Right-click** on the **plb_fish_finder** pcore and **select Add IP**. You will see the pcore appear in the **System Assembly View** as shown below:



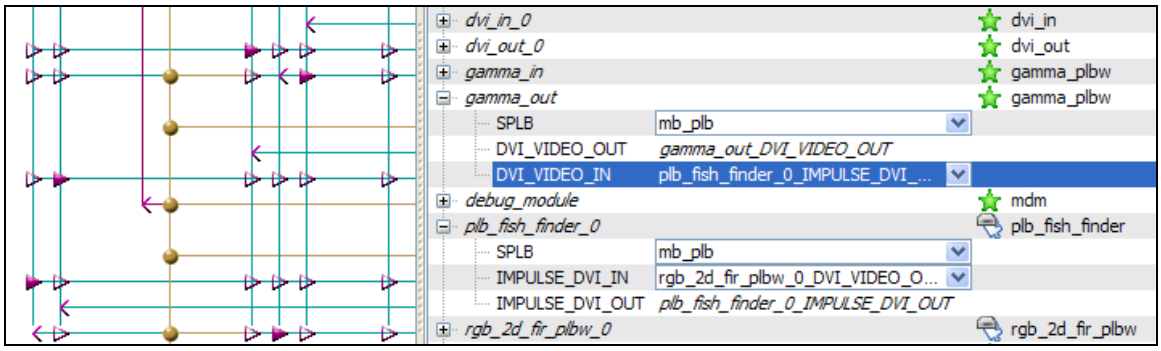
2. The pcore must now be attached to the processor local bus, and must also be connected to the **RGB 2D** and **Gamma Out** pcores via video bus interfaces.

   a. In the System Assembly View expand **plb_fish_finder** by **clicking** on the small ⊞ icon to view the available connections:

   b. Connect the **plb_fish_finder_0** filter to the PLB by **clicking** on the small open circle to the left of the SPLB entry. This will establish the **plb_fish_finder_0** as a slave on the PLB.

   c. Connect the DVI input to the **plb_fish_finder** filter by **clicking** on the **IMPULSE_DVI_IN** drop-down list, and **select rgb_2d_fir_plbw_0_DVI_VIDEO_OUT.**

d. Connect the DVI output of the **plb_fish_finder** filter to the Gamma Out pcore by **clicking** on the small ⊞ icon next to the **gamma_out** pcore entry, then **click** on the **DVI_VIDEO_IN** drop-down list and **select plb_fish_finder_0_IMPULSE_DVI_OUT.**
The connections should appear as follows:



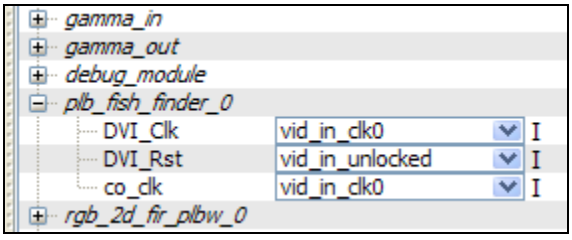
3. The **plb_fish_finder_0** component has now been attached to the processor local bus (PLB) and to the video pipeline via the DVI interfaces. Next, add clock and reset signals to the pcore.

   a. To connect the **plb_fish_finder** clock and reset signals, **select** the **Ports** tab of the **System Assembly View**.

   b. Find the entry for **plb_fish_finder_0** in the list of pcores, and **click** the small ⊞ icon to expand the entry.

   c. For the **co_clock** AND **DVI_Clk** connections; use the drop-down selection list to **select vid_in_clk0** for each.

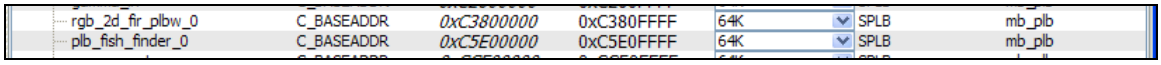
   d. For the **DVI_Rst** connection; use the **drop-down** selection list to **select** the **vid_in_unlocked** signal.



4. Next create the base address for the added pcore

   a. **Click** on the **Addresses** tab in the **System Assembly View**

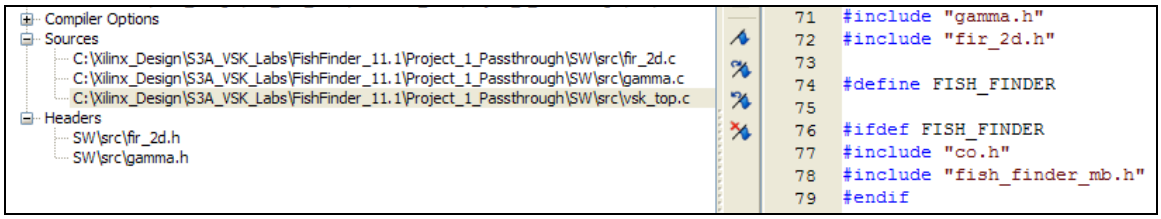
   b. **Click** the **Generate Addresses** button to automatically assign a base address and Size to the **plb_fish_finder_0** pcore.



Part 5 – Modify the MicroBlaze Software Application

The hardware portion of the application is now complete and ready to generate as an FPGA bitmap. Before doing so, however, we will need to make some modifications to the software portion of the application to add in a new menu. This menu will allow us to control aspects of the object detection filter, including the type of fish to highlight, and the size of the spotlight.

1. Modify the software application as follows:
   a. **Select** the **Applications** tab of the **Project Information Area**.
   b. **Click** on the small ⊞ icons next to the **Sources** and **Headers** entries to expand and view the files.
   c. Open the **vsk_top.c** file by **double-clicking** on the file name to invoke the text editor.
   d. Scroll down to near line 74 in the file, and locate the lines defining **FISH_FINDER**. Uncomment this line by removing the leading // characters (including the leading space character). This will enable an additional menu option in the MicroBlaze software application
   e. **Select** on the **Menu Bar: File ⇒ Save** to save the modified file



2. To include the additional "fish finder" control files to the MicroBlaze software application, add two additional C source files and two header files to the project.
   a. **Right-click** on the **Sources** entry and selecting the menu item **Add Existing Files**. When prompted for the file names, navigate to:

   c:\xilinx_design\S3A_VSK_Labs\FishFinder
   Project_1_Passthrough_Completed\code

   and **select** the two C source files:

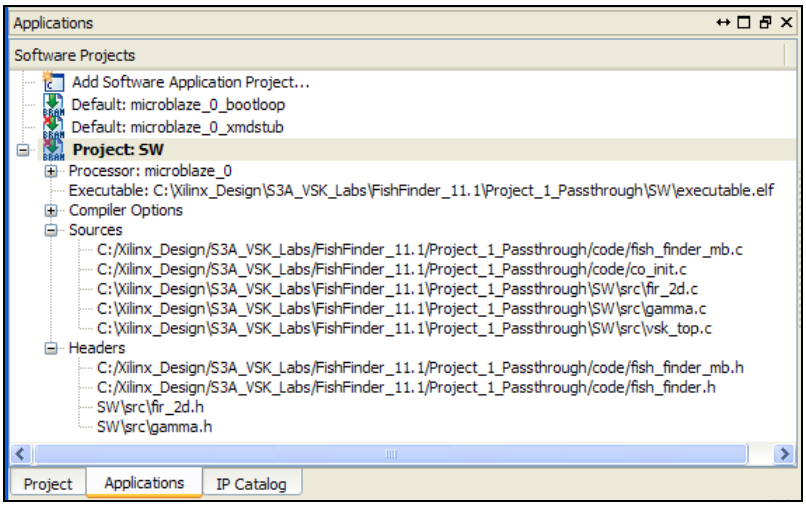
   co_init.c **fish_finder_mb.c**

   b. **Click** the **Open** button to add the files to the project.
   c. **Right-click** on the **Headers** entry and **select Add Existing Files**. Navigate again to the **code** directory, and **select** the two ".h" files:

   fish_finder.h **fish_finder_mb.h**

d. **Click** the **Open** button to add these files to the project.

e. The **Software Projects** Information Area should now appear as follows:



3. The project is now ready to be built. To save time and skip the hardware build step, go directly to a pre-built design as follows:

   a. **Select** on the **Menu Bar:**
   File ⇒ Open Project

   b. Browse to and **select** the following file:

   c:\xilinx_design\S3A_VSK_Labs\FishFinder
   Project_2_Fish_Finder_Completed\system.xmp

   Hint: The desired directory should be up one level from the current directory.

   c. Click Open

   d. If a pop-up window prompting if it is OK to close the existing project appears, **Click YES**

   e. Select on the Menu Bar:
   Software ⇒ Build All User Applications

   f. A series of build steps will complete automatically, concluding with the compilation and linking of the MicroBlaze executable file, **executable.elf**.

You have now completed creating the hardware and software system!

Part 6 – Build and Run the Fish Finder Design

1. Build, download and run the software design

   a. **Select** on the **Menu Bar**:

      Device Configuration ⇒ Download Bitstream

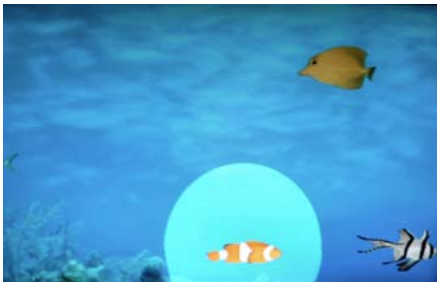
   b. To download and run the software design, Select on the Menu BAR:

      Debug ⇒ Launch XMD

      Note: We have modified the XMD command file to automatically download and run the software.

   c. Close the XMD command window by **typing** "**exit"** at the prompt.

   d. At this point you should see the video appearing on the output display monitor and observe that the clown fish in the sample video is being detected and highlighted as shown.



   e. Buttons on the VSK board are read by the MicroBlaze processor and may be used to change the fish detected and the spotlight size:

      - East Button - Find Yellow Fish
      - Center Button - Find Clown Fish
      - West Button - Find Famous Fish
      - North Button - Increase Spotlight
      - South Button - Decrease Spotlight

   f. **Pressing** the **East** and **West** Buttons will enable the HyperTerminal menu to have the MicroBlaze control the fish detected and the spotlight size:

      - **Press** the **Enter** key to see the main menu
      - **Press** the **'n'** key to display the Fish Finder menu
      - **Select** the desired change



You have now completed the Fish Finder video application!