AN 045: Using the Video Affine Transformation Example Design

AN045-v1.0
May 2022
www.efinixinc.com

Contents

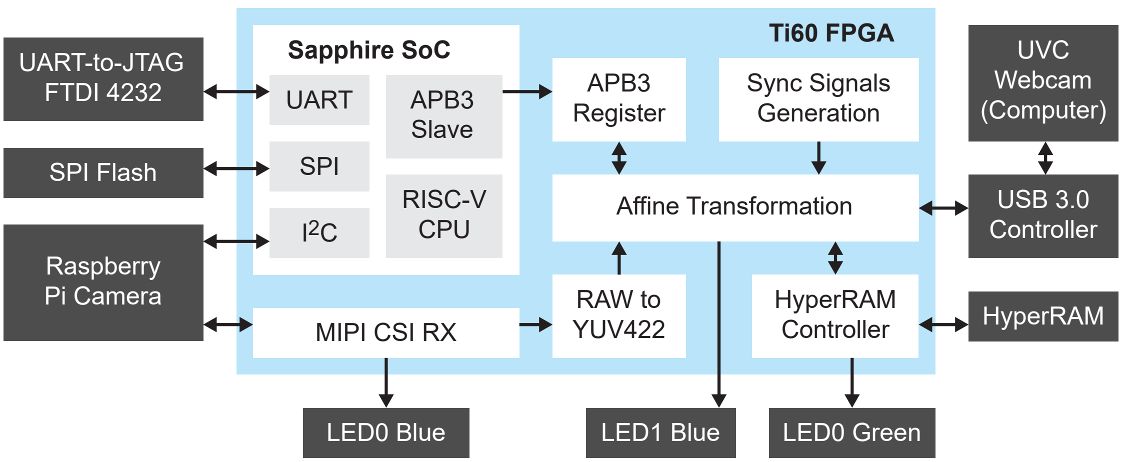
Introduction

Efinix provides a video affine transformation example design that demonstrates how Titanium Ti60 FPGA can process real-time video affine transformation processing using the Titanium Ti60 F225 Development Board. The design supports rotate, scale, translate, and reflect transformations.

The design receives video input from the Raspberry Pi camera module, apply transformation, then outputs the video stream to the Windows Camera app through USB 3.0 interface. The design uses the YUV422 data format and supports up to 1920 x 1080 at 30 fps. The design includes a Sapphire RISC-V SoC that generates the affine transformation matrix based on the design parameters. The Ti60 FPGA then uses the this matrix to perform the transformation. You can adjust the transformation parameters without the need to recompile your design.

Efinix also includes an additional example design with the similar functions that can be used with the Google Coral camera instead of the Raspberry Pi camera module.

Figure 1: Video Affine Transformation Example Design Block Diagram



Titanium Design Implementation

Example with 1920 x 1080 input resolution, 1080 x 1920 output resolution, and 30 frame per second.

FPGA	Design Module	eXchangeable Logic and Routing (XLR) Cells	Memory Blocks	DSP48 Blocks	Efinity Version
Ti60 F225 C4	Entire Design	15,193	193	22	2021.2
	Affine Transformation Module Only	3,612	125	10	

Required Hardware

The example design uses the following hardware from the Titanium Ti60 F225 Development Kit:

- Titanium Ti60 F225 Development Board
- 1 Dual Raspberry Pi Camera Connector Daughter Card
- 1 Raspberry Pi v2 camera module
- 1 15-pin flat cable
- 2 USB type-C cable
- Universal AC to DC power adapter

The following optional hardware are required for the example design using the Google Coral camera:

- 1 Google Camera module
- 1 Google Camera Connector Daughter Card
- 1 24-pin, 0.5 mm pitch, opposite side flat cable

Required Software

The example design uses the following software:

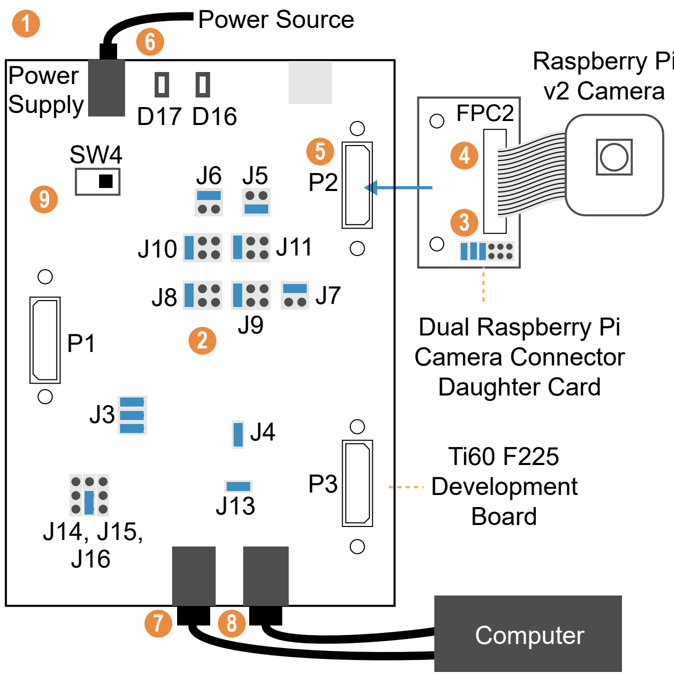
- Windows 10 Camera App
- Terminal program such as Putty, termite or the built-in Eclipse terminal

Note: You need to install the RISC-V SDK if you plan to modify the Sapphire SoC application in this example. Refer to the **Sapphire RISC-V SoC Hardware and Software User Guide** for detailed information about installing and setting-up the RISC-V SDK for Sapphire RISC-V SoC.

Set Up the Hardware

The following figure shows the hardware setup steps. If you have not already done so, attach standoffs to the board.

Figure 2: Hardware Setup



1. Attach standoffs to the board if you have not already done so.
2. On the Titanium Ti60 F225 Development Board, connect the following jumpers:

Header	Net Name	State	Short Pins
J3	VCC	0.95V	1 and 2
J3	VCC	0.95V	3 and 4
J3	VCC	0.95V	5 and 6
J4	VCCAUX	1.8V	1 and 2
J5	VCCIO33_TR	3.3V	1 and 2
J6	VCCIO33_TL	1.8V	3 and 4
J7	VCCIO33_BR	1.8V	3 and 4
J8	VCCIO3A	1.2V	5 and 6
J9	VCCIO3B	1.2V	5 and 6
J10	VCCIO4A	1.2V	5 and 6
J11	VCCIO4B	1.2V	5 and 6
J13	SPI_ENA	Short	1 and 2
J14	FX3_PMODE0	Float	Unconnected
J15	FX3_PMODE1	Pull-up	1 and 2
J16	FX3_PMODE2	Float	Unconnected

3. On the Raspberry Pi Camera Connector Daughter Card, connect the following pins with jumpers: 1 - 2, 3 - 4, and 5 - 6.
4. Connect the Raspberry Pi v2 camera module to the FPC2 connector of the Raspberry Pi Camera Connector Daughter Card using the 15-pin flat cable.
5. Connect the Raspberry Pi Camera Connector Daughter Card to the P2 header of the Titanium Ti60 F225 Development Board.
6. Ensure the board power switch is turned off, then connect the 12 V power cable to the board connector and to a power source.
7. Connect the USB header J1 to a USB 3.0 port of your computer.
8. Connect the USB header J12 to USB port of your computer.
9. Turn on the board's power switch.

Program the Titanium Ti60 F225 Development Board

The Titanium Ti60 F225 Development Board ships pre-loaded with an example design. To use the Video Affine Transformation example design, you must program the design into the board.

Note: You need to program the USB 3.0 controller if you have programmed it with a different design. Refer to the USB 3.0 Controller section of the **Titanium Ti60 F225 Development Kit User Guide** for instructions on how to program the USB 3.0 controller. Program the USB 3.0 controller first before you proceed with the following steps.

1. Download the example design file, **video-affine-transform-v<version>.zip**, from the Support Center.
2. Open the project (**affine_transform.xml**) in the Efinity® software and review it. The project is located in the **hw/efinity/affine_transform_piv2** directory.
3. Use the Efinity® Programmer and SPI active mode to download the bitstream file to your board. The example includes a bitstream file, **affine_transform_w_riscv.hex**. You use SPI active mode because you need to reset the FPGA.
4. Press SW3 (CRESET) on the Titanium Ti60 F225 Development Board to reset the FPGA or turn the board power on and off.

After the FPGA configures, the board LEDs light up to indicate the following status:

LED	Light Status	Description
D16	Green light turn on	The HyperRAM calibration is successful.
D16	Blue light blinking	The board is receiving video data from the camera.
D17	Blue light blinking	The USB 3.0 controller is receiving data from the affine transformation module.

Run the Example Design

Setting-Up

1. Open **Windows Device Manager**, under **Camera**, you should see **Ti60** device in the list. Turn the development board off and on if it is not listed.
2. Open the Camera app in Windows 10. The Camera app displays the video captured by the Raspberry Pi Camera with a 90-degree clockwise rotation.
3. Open a terminal to connect to the UART. You can use any terminal program, such as Putty, termite, or the built-in Eclipse and enter the following settings:

Option	Setting
Choose terminal	Serial Terminal
Serial port	COM*n* where *n* is the port number for your UART module. The UART port is usually the first device under **Ports (COM & LPT)** in the **Windows Device Manager**.
Baud rate	115200
Data size	8
Parity	None
Stop bits	1
Encoding	Default (ISO-8859-1)

Applying Video Transformations

The terminal prints `low speed error count: 0` continuously when the design is running correctly. Type the following commands in the terminal to apply affine transformations:

Table 1: Design Example Video Transformation Commands

Transformation	Command	Description
Rotation	angle <*value*>	Apply rotation to the video by <*value*> degrees. Positive value: Counter clockwise rotation Negative value: Clockwise rotation
	center_x <*value*> center_y <*value*>	Set the rotation center point coordinates (in pixels) for x-axis and y-axis. The origin, (0,0), is at the top left corner of the input video. Supports positive and negative values.
Scale	scale <*value*>	Apply scaling to the video. The scaling factor is <*value*>/10000.
Translation	translate_x <*value*> translate_y <*value*>	Apply translation to the video along the x-axis and y-axis. Supports positive and negative.
Reflection	reflect_x <*value*> reflect_y <*value*>	Apply reflection over x-axis and y-axis. 1 : Enable reflection 0 : Disable reflection The image may be out of the display area after reflection. You can use translation to move the image back into the display area.

Revision History

Table 2: Revision History

Date	Version	Description
May 2022	1.0	Initial release.