

# **Colour Block Detection**

## **Custom IP**

**ECE532**

**Kazi Sudipto Arif  
Shariq Khalil Ahmed**

# Overview

## Functional Description

The Colour Block Detection IP is a block that can detect a continuous object of colour from video input given that it meets a specific colour threshold (see Figure 1-1). It outputs the x and y coordinates of the square enclosing this object so that it can be processed for whatever purpose the designer wants (i.e. use the coordinates of that object on the screen for image manipulation). The colour thresholds have to conform to 8-bits per channel RGB specification for pixel detection meaning that each pixel is 24 bits. The finite state machine used to detect the colored object simply detects the first pixel matching the threshold in each line and updates the border coordinates until the threshold is not met in that line. If the next line has no pixels meeting the threshold, detection is finished. Otherwise the border coordinates are updated for this line and it keeps checking each line till screen refreshes. When the screen refreshes, the coordinates of the object are output.



Figure 1-1: Detection of a green object in demo (note the lines surrounding it)

The Colour Detection Block has a single AXI Lite slave interface, a single AXI Stream slave interface, and a single AXI Stream master interface (see Figure 1-2) . The video is streamed to the slave interface and passes out the other side unaltered via the master interface. The Colour Detection Block reads every pixel that is passed through.

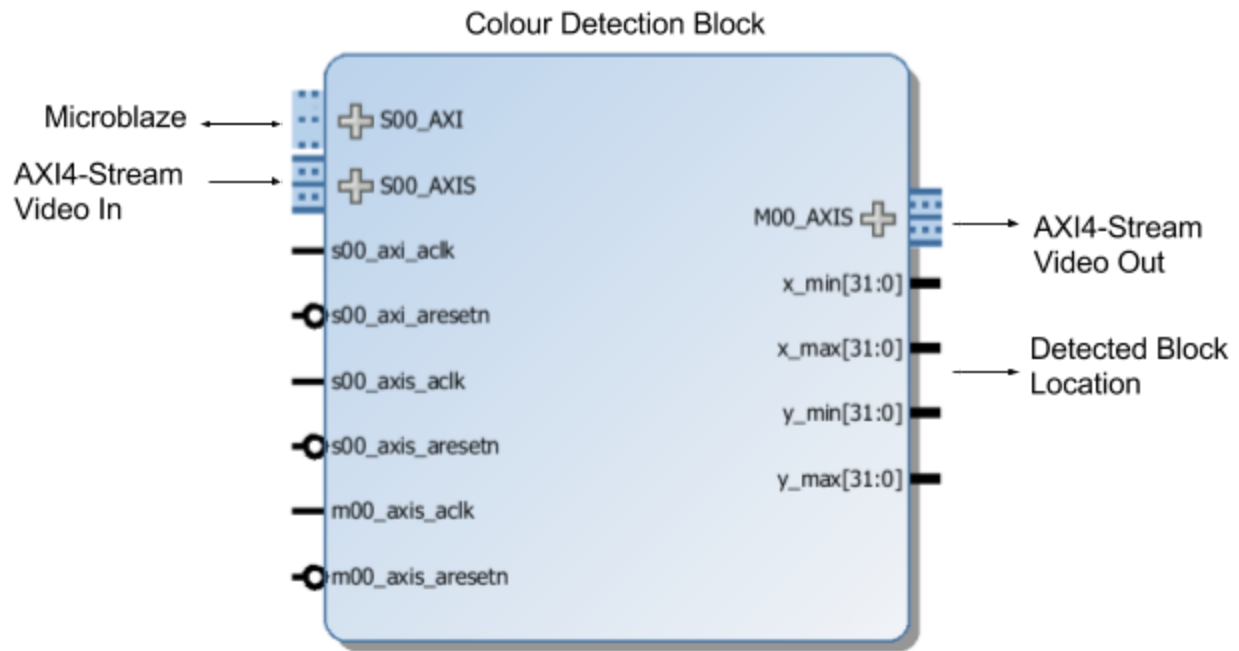


Figure 1-2: Colour Detection IP Block

### *AXI4-Lite Interface*

The AXI4-Lite Interface module implements a 32-bit AXI4-Lite slave interface for accessing 10 slave registers.

### *AXI4-Stream Interface*

The peripheral has a slave-to-master AXI4-Stream interface (input video stream) and a master-to-slave AXI4-Stream interface (output video stream).

### *Colour Block Detection Core*

The Colour Block Detection Core consists of logic writing to each slave register. Furthermore it includes logic to detect a block of colour via the pixels being streamed in and outputs the x and y coordinates for the box surrounding this object. The x and y coordinates are output once the entire screen has been read in via video.

## Product Specification

Performance characterization of the IP was done using a 100Mhz clock on an Artix-7 (XC7A200T-1SBG484C) on the Nexys 4 Video board. The resource utilization on the chip is in Table 2-1.

Table 2-1: Device Resource Utilization for AXI PRAND\_NUM\_GEN slave on Artix-7

Slices	Flip-Flops	LUTs
254	922	535

### *Port Descriptions*

The Colour Block Detection IP I/O signals are listed and described in Table 2-2. The interface for the coordinates of the object can be connected to any subsequent block that might need to process the coordinates. The initial state for the minimum values is a large number and 0 for the maximum values as these allow comparing new minimum and maximum values in the finite state machine respectively.

Table 2-2: Colour Block Detection I/O Signal Description

Signal Name	Interface	I/O	Initial State	Description
s_axi_aclk	Clock	I		AXI Clock.
s_axi_aresetn	Reset	I		AXI Reset, active-Low.
s_axi_*	S_AXI	NA	-	AXI4-Lite Slave Interface signals.
s_axis_aclk	S_AXIS	I		AXIS Clock, should be tied to the clock driving pixels
s_axis_aresetn	S_AXIS	I		AXIS Reset, active-Low, not used by this block
s_axis_*	S_AXIS	NA	-	AXI4-Stream Slave Interface signals
m_axis_aclk	M_AXIS	I		AXIS Clock, should be tied to the clock driving pixels
m_axis_aresetn	M_AXIS	I		AXIS Reset, active-Low, not used by this block

m_axis_*	M_AXIS	NA	-	AXI4-Stream Master Interface signals
x_min_out	Any Block	O	0xffffffff	32-bit unsigned int signifying left border of detected region on x-axis
x_max_out	Any Block	O	0	32-bit unsigned int signifying right border of detected region on x-axis
y_min_out	Any Block	O	0xffffffff	32-bit unsigned int signifying top border of detected region on y-axis
y_max_out	Any Block	O	0	32-bit unsigned int signifying bottom border of detected region on y-axis

## Register Space

Table 2-3 shows the registers and their addresses.

**Table 2-3: Registers**

Address Space Offset	Register Name	Access Type	Data Type	Default Value	Description
0x0000	R_MIN	R/W	8-bit unsigned int	100	Red Minimum Threshold
0x0004	R_MAX	R/W	8-bit unsigned int	255	Red Maximum Threshold
0x0008	G_MIN	R/W	8-bit unsigned int	100	Green Minimum Threshold
0x000c	G_MAX	R/W	8-bit unsigned int	255	Green Maximum Threshold
0x0010	B_MIN	R/W	8-bit unsigned	100	Blue Maximum Threshold

			int		
0x0014	B_MAX	R/W	8-bit unsigned int	255	Blue Maximum Threshold
0x0018	X_MIN	R	32-bit unsigned int	0xffffffff	Minimum x-value for detected region
0x001c	X_MAX	R	32-bit unsigned int	0x0	Maximum x-value for detected region
0x0020	Y_MIN	R	32-bit unsigned int	0xffffffff	Minimum y-value for detected region
0x0024	Y_MAX	R	32-bit unsigned int	0x0	Maximum y-value for detected region

All registers are available for use in all configurations of the colour Block Detection IP.

### *Colour Threshold Registers Overview*

All the slave registers for this block are used to set up the colour range that the object that is being detected should fall in. A range is required due to noise and lighting affecting pixel values and the minimum range has to be smaller than the maximum range or else nothing will be detected. By default the block detects whites that fall in the range of 100-255 for each of the RGB channels. The color range for each of the three channels in RGB is 0 to 255.

### *Block Coordinate Registers Overview*

The block coordinate registers can be read and used if software image manipulation is desired instead. This means reading the coordinate registers and using the values.

## **Designing with the Core**

### *Operation*

The Colour Detection Block requires a video input in RGB format being streamed into the AXIS slave interface. The AXIS slave interface clock input needs to be tied to the clock driving each pixel. The block is always running and attempting to detect an object of the specified colour from video input. The master interface can be connected to a video output or a cascade of Colour Detection Block IPs if multiple object detection is required. The x and y coordinates of the object can be output to another block if further processing is required. If hardware manipulation of these coordinates is not desired then software manipulation can take place by reading the coordinate registers.

## *Programming Sequence*

The programmer simply needs to write the values of the object's colour threshold to the six slave registers and the block will start detecting that colour threshold. The detected region coordinates can be read any time as well.

## *Clocking*

The Colour Block Detection IP operates on the s\_axi\_clk. The stream interfaces can operate on the stream source clocks.

## *Resets*

When s\_axi\_resetn is asserted, the entire block resets including the detection FSM. This is an active-low reset synchronous to s\_axi\_clk clock.