UG0641

User Guide

Alpha Blending

February 2018

Power Matters.™

Contents

1 Revision History 1
1.1 Revision 4.0 1
1.2 Revision 3.0 1
1.3 Revision 2.0 1
1.4 Revision 1.0 1

2 Introducing Alpha Blending 2

3 Hardware Implementation 3
3.1 Inputs and Outputs 3
3.2 Configuration Parameters 5
3.3 Timing Diagrams 5
3.4 Testbench 6
3.5 Resource Utilization 13

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 4.0

In revision 4.0 of this document, the Resource Utilization section and the Resource Utilization Report table were updated. For more information, see Resource Utilization (see page 13).

1.2 Revision 3.0

In revision 3.0 of this document, the Testbench section was updated with the Steps to simulate the core using test bench. For more information, see Testbench (see page 6).

1.3 Revision 2.0

In revision 2.0 of this document, the SAR 76066 was updated. For more information, see Testbench (see page 6).

1.4 Revision 1.0

Revision 1.0 is the first publication of this document.

2 Introducing Alpha Blending

Alpha blending is the process of combining an image with a background to create the appearance of partial or full transparency. It is used to render multiple images into a single background image in separate passes and make one final image. Alpha blending can be done in two different methods:

- Pixel-wise alpha blending
- Global alpha blending

In pixel-wise alpha blending, in addition to red, green, and blue (RGB) color channel, each pixel has an additional alpha channel. The alpha channel contains as many bits as a color channel. For example, an 8-bit alpha channel can represent 256 levels of transparency with a value of 0 denoting that complete image is transparent and a value of 255 denoting that the complete image is opaque. This alpha blending type defines the transparency of individual images when blended with the background image.

The following is the equation for pixel-wise alpha blending:

$V_{Out} = RGB_{Image1} \times \alpha_{aImage1} + RGB_{BGImage} \times (1 - \alpha_{aImage1})$

Where,

RGB_{Image1} = RGB values of Image 1

α_{Image1} = Alpha pixel value of Image 1

$RGB_{BGImage}$ = RGB value of background image

V_{Out} = Resultant alpha-blended RGB value

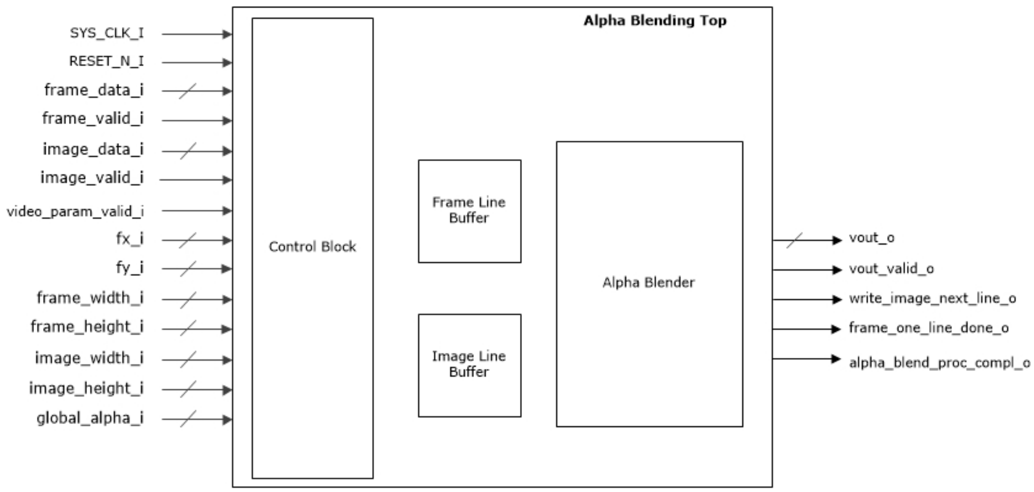
In case of global alpha blending, a global alpha value is applied uniformly on all pixels of image. This type of alpha blending defines the overall transparency of the image.

$V_{Out} = RGB_{Image} \times \alpha_{aImage}$

3 Hardware Implementation

The following figure shows the block diagram of alpha blending.

Figure 1 • Block Diagram of Alpha Blending Block



3.1 Inputs and Outputs

The following table lists the description of input and output ports.

Table 1 • Inputs and Outputs of Alpha Blending Block

Signal Name	Direction	Width	Description
RESET_n_I	Input	-	Active low asynchronous reset signal to design
SYS_CLK_I	Input	-	System clock
frame_data_i	Input	[g_FRAME_DATAWIDTH-1:0]	Frame pixel data input
frame_valid_i	Input	-	Frame input data valid
image_data_i	Input	[g_IMAGE_DATAWIDTH-1:0]	Image pixel data input with most significant bit (MSB)s representing alpha value
image_valid_i	Input	-	Image input data valid
video_param_valid_i	Input	-	Valid input for sampling video parameters like Fx, Fy, frame width and height, and image width and height
fx_i	Input	[g_FRAME_X_Y_DATAWIDTH-1:0]	Valid input for sampling video parameters like Fx, Fy, frame width and height, and image width and height
fy_i	Input	[g_FRAME_X_Y_DATAWIDTH-1:0]	Vertical Y location on the background frame where the image has to be alpha-blended. Top left Y location on the background frame.
frame_width_i	Input	[g_FRAME_X_Y_DATAWIDTH-1:0]	Width of the background frame
frame_height_i	Input	[g_FRAME_X_Y_DATAWIDTH-1:0]	Height of the background frame

Signal Name	Direction	Width	Description
image_width_i	Input	[g_IMAGE_X_Y_DATAWIDTH-1:0]	Width of the image
image_height_i	Input	[g_IMAGE_X_Y_DATAWIDTH-1:0]	Height of the image
global_alpha_i	Input	[(g_IMAGE_DATAWIDTH/4)-1:0]	Global alpha value to be applied on complete frame
vout_o	Output	[g_OUTPUT_CHANNEL_DATAWIDTH-1:0]	Alpha blended frame output data
vout_valid_o	Output	-	Output data valid
write_image_next_line_o	Output	-	Output to external controllers to write the next image line into the inline buffer. An active high pulse of one clock cycle width is generated.
frame_one_line_done_o	Output	-	One line completion for the frame. An active high pulse of one clock cycle width is generated.
alpha_blend_proc_compl_o	Output	-	Completion of the alpha blending process for one complete frame. An active high pulse of one clock cycle width is generated.

The alpha blending block can blend the image pixel data into the frame pixel data depending on the alpha channel value, which is part of image pixel. The image pixel data structure with 8-bit width for each channel is listed in the following table.

Table 2 • Image Pixel Data Structure

Bit	Channel
[31:24]	Alpha
[23:16]	Red
[15:8]	Green
[7:0]	Blue

The alpha value from the image pixel along with RGB values of image and frame is used to calculate the output alpha blended pixel value, refer to EQ1. In addition, the block can apply a global alpha on the output frame using the global_alpha_i input.

The block along with the image and frame pixel data inputs, it also takes frame height and width (frame_width_i, frame_height_i), image height and width (image_width_i, image_height_i), and horizontal X and vertical Y positions (fx_i, fy_i) as inputs. These inputs are used by the control block to define the region of the alpha blending on the frame image. Therefore, external controllers can specify dynamic region for alpha blending the image for each input frame. The alpha blending location, image, and frame size input parameters are sampled by the block when video_param_valid_i input is toggled. These parameters must be constant for one complete frame. Completion of alpha blending for one complete frame is indicated by toggling of alpha_blend_proc_compl_o output. Before writing the first line of frame and image pixels, the alpha blending location and image and frame size inputs have to be provided to the block.

The alpha blending block has one line of buffers each for image and frame. Size of these two buffers can be modified using configuration parameters depending on the maximum line widths of input images and frames. To facilitate the writing of image and frame lines into the inline buffers, the block generates write_image_next_line_o and frame_one_line_done_o outputs for external controllers. The image data must be written into the respective inline buffer followed by frame data into its inline buffer. When five pixels of the frame data is written into the frame inline buffer, the alpha blending module starts by itself and performs the alpha blending process depending on the location and image and frame size input parameters.

3.2 Configuration Parameters

The following table lists the description of the configuration parameters used in the hardware implementation of Alpha Blending block. They are the generic parameters and can vary based on the application requirements.

Table 3 • Configuration Parameters

Name	Description
g_FRAME_DATAWIDTH	RGB data bit width for the Frame channel
g_IMAGE_DATAWIDTH	ARGB data bit width for image channel
g_FRAME_X_Y_DATAWIDTH	Frame height, width, and X and Y location data bit width
g_IMAGE_X_Y_DATAWIDTH	Image height and width data bit width
g_FRAME_BUFFER_AWIDTH	Frame inline buffer address bit width
g_IMAGE_BUFFER_AWIDTH	Image inline buffer address bit width
g_OUTPUT_CHANNEL_DATAWIDTH	RGB data bit width for the output channel

3.3 Timing Diagrams

The following figure shows the alpha blending pixel output in connection with the frame pixel input. The alpha blending process starts once five pixels of the frame are written into the frame inline buffer. After the process starts, it takes eight clock cycles for first output pixel.

Figure 2 • Timing Diagram showing Frame Data Input to Output Data

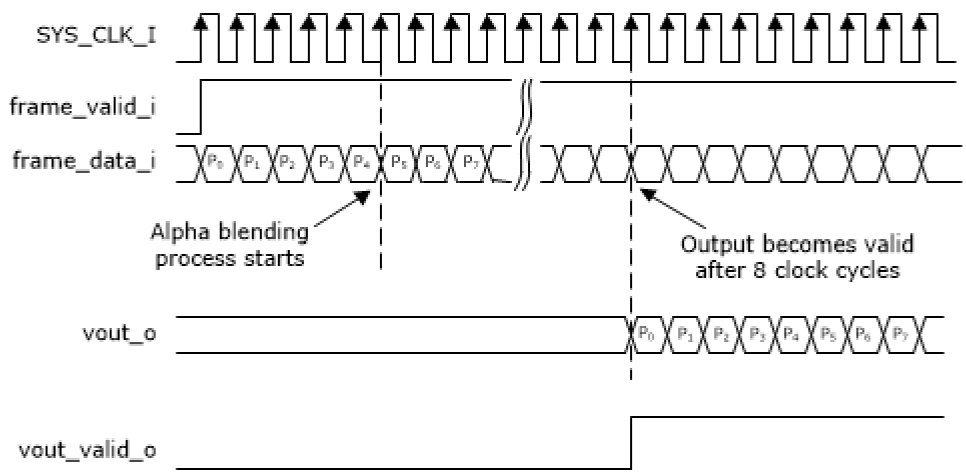


Figure 3 • Timing Diagram showing Frame Data Input with Corresponding Data Valid

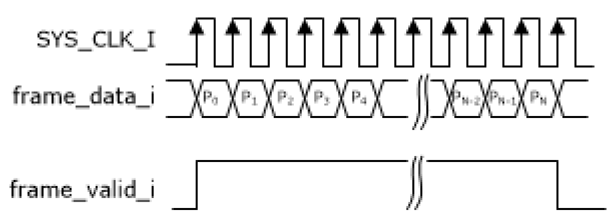


Figure 4 • Timing Diagram showing Image Data Input with Corresponding Data Valid

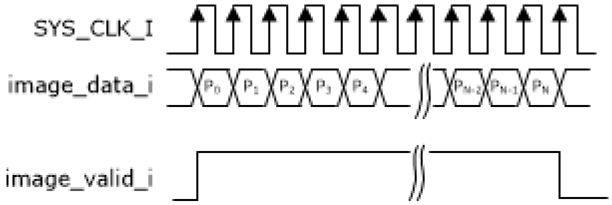
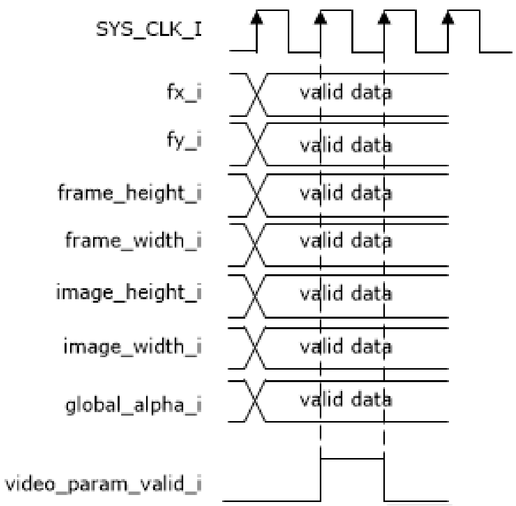


Figure 5 • Timing Diagram showing Video Parameters with Valid Input



3.4 Testbench

A testbench is provided to check the functionality of alpha blending core. The following table lists the parameters that can be configured according to application.

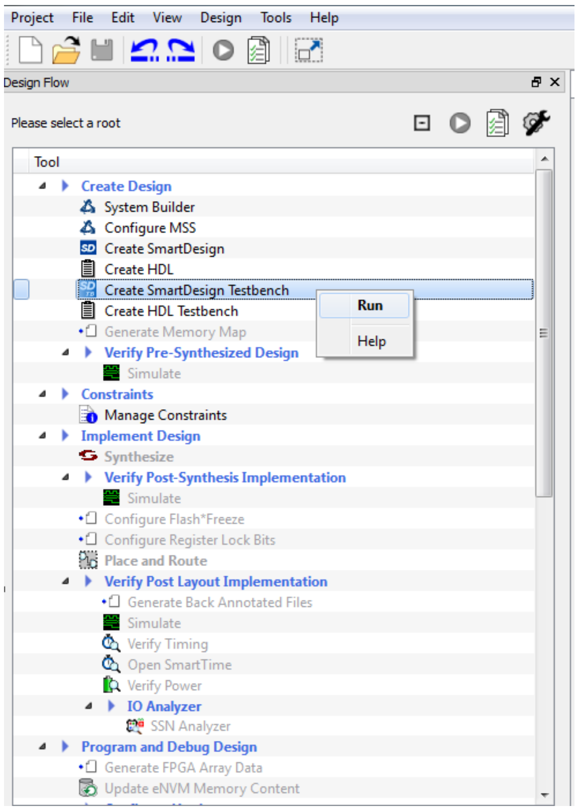
Table 4 • Testbench Configuration Parameters

Name	Description
CLKPERIOD	Clock Period
IMAGE_HEIGHT	Height of the image
IMAGE_WIDTH	Width of the image
FRAME_WIDTH	Width of the frame
FRAME_HEIGHT	Height of the frame
IMAGE_FILE_NAME	Input image name

The following steps describe how to simulate the core using the testbench.

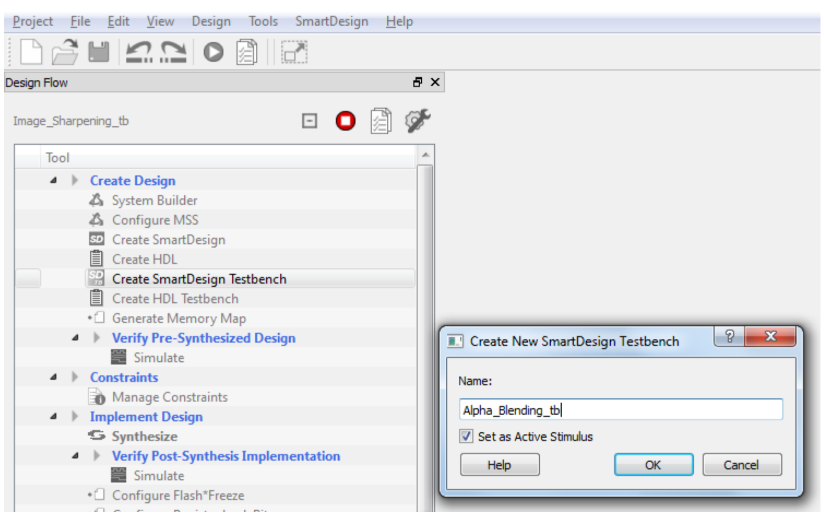
1. In the **Design Flow** window, expand **Create Design**. Right-click **Create SmartDesign Testbench** and click **Run** as shown in the following figure.

Figure 6 • Create SmartDesign Testbench



2. Enter a name for the SmartDesign testbench, and click **OK**.

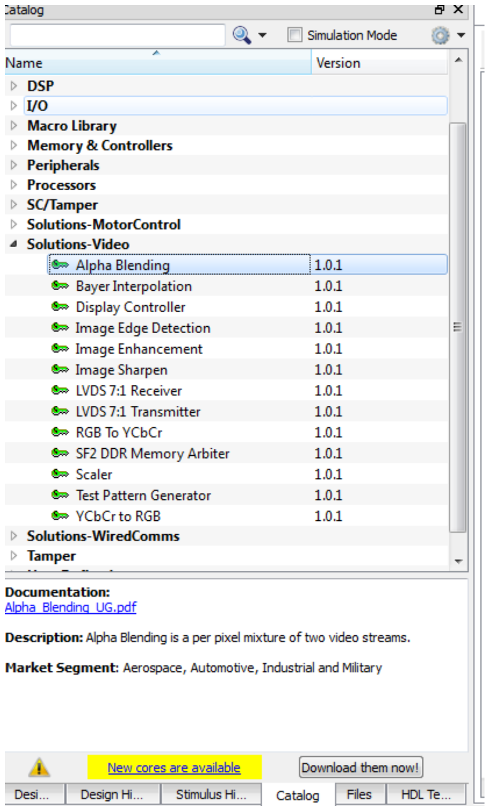
Figure 7 • Create New SmartDesign Testbench Dialog



A SmartDesign testbench is created, and a canvas appears to the right of the **Design Flow** pane.

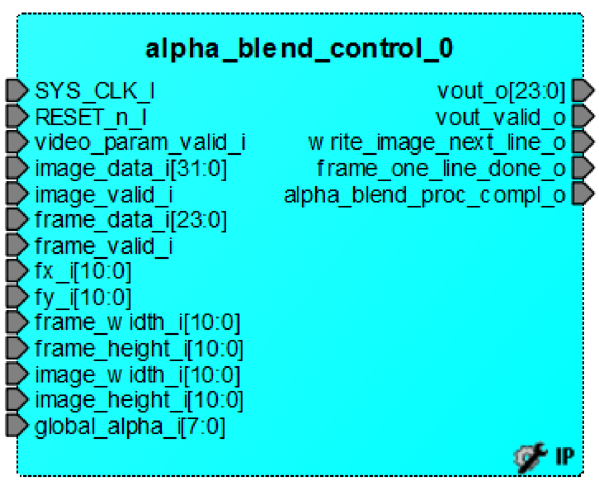
3. In the **Catalog** window, expand **Solutions-Video**, and drag the **Alpha Blending** onto the **SmartDesign testbench** canvas.

Figure 8 • Alpha Blending



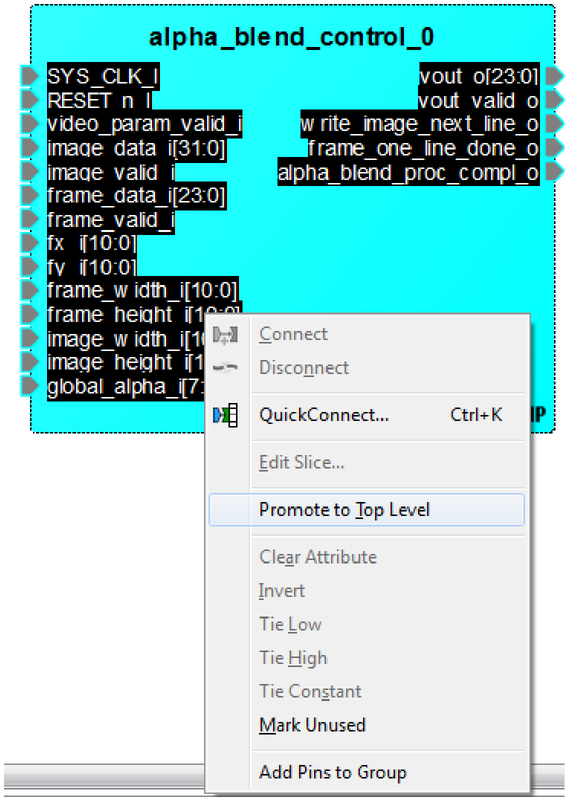
The core appears on the canvas as shown in the following figure.

Figure 9 • Alpha Blending Core on SmartDesign Testbench Canvas



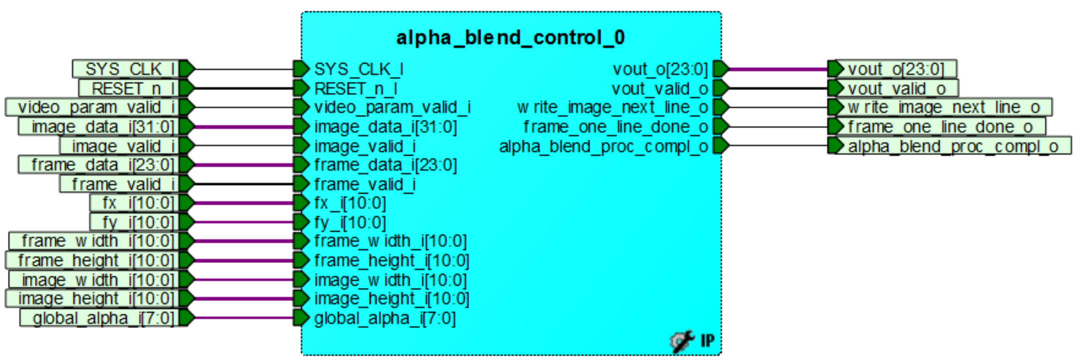
4. Select all the ports of the core, right-click, and select **Promote to Top Level** as shown in the following figure.

Figure 10 • Promote to Top Level



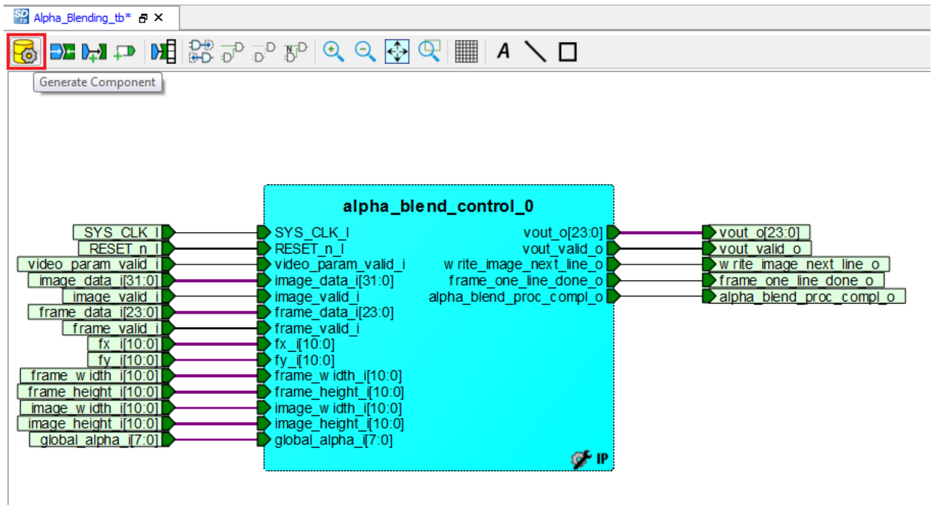
The ports are promoted to the top-level as shown in the following figure.

Figure 11 • Alpha Blending Promoted to Top Level



5. Click **Generate Component** from the SmartDesign toolbar as shown in the following figure.

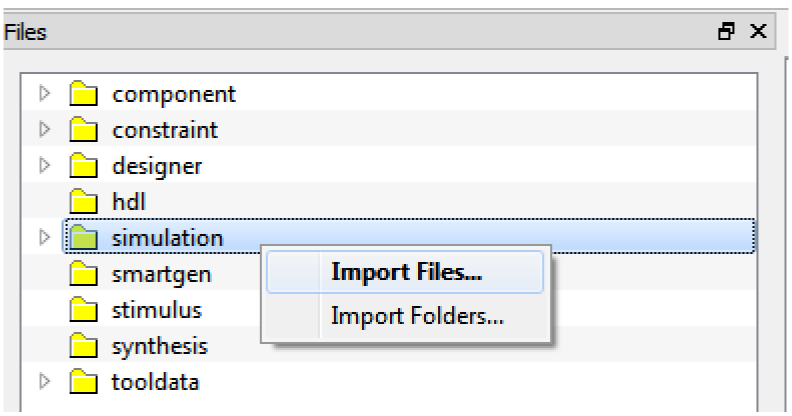
Figure 12 • Generate Component Icon



A sample image is available in the Stimulus hierarchy.

6. In the **Files** window, Right-click **simulation** and click **Import Files...** as shown in the following figure.

Figure 13 • Import Files Option



7. Do one of the following:
   - To import the sample testbench input image, browse to sample testbench input file to the stimulus directory, and click Open as shown in the following figure.
   - To import a different image, browse to the folder containing the image file, and click **Open**.

8. Import this file into the Simulation folder in the Files folder.

Figure 14 • Input Image File Selection

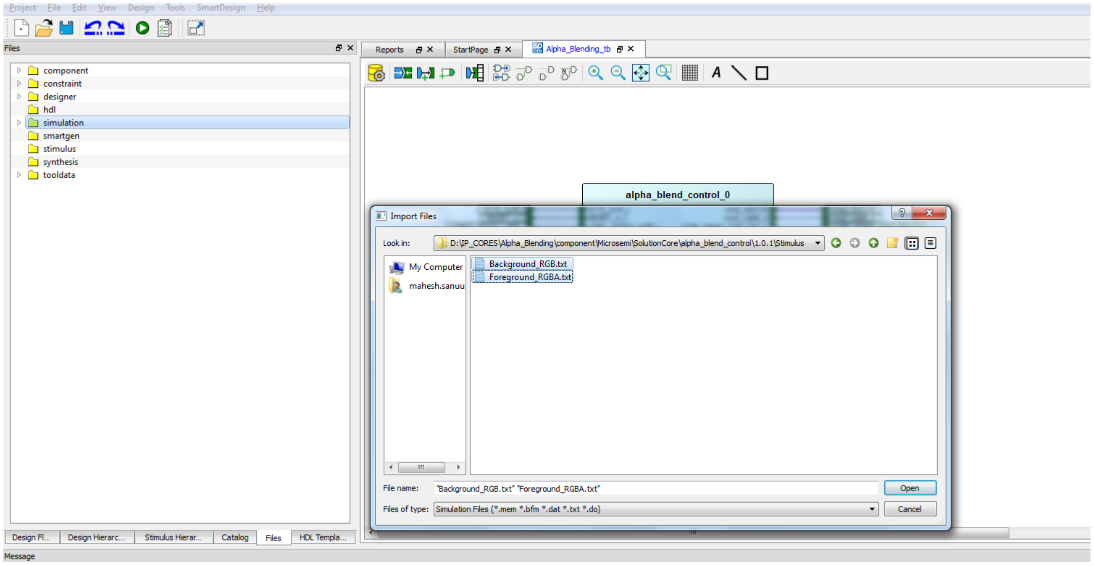
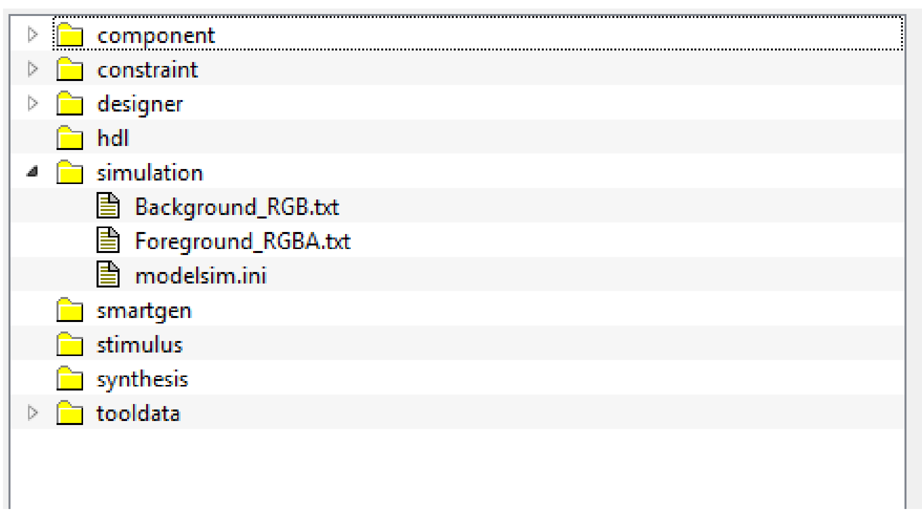
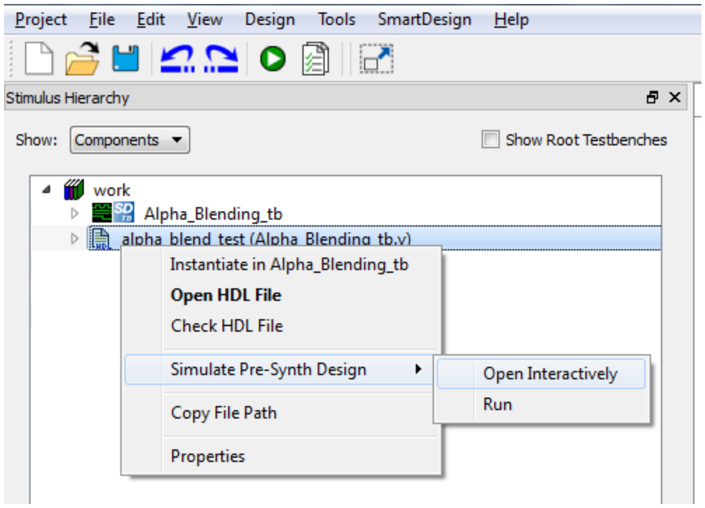


Figure 15 • Input Image File in Simulation Directory



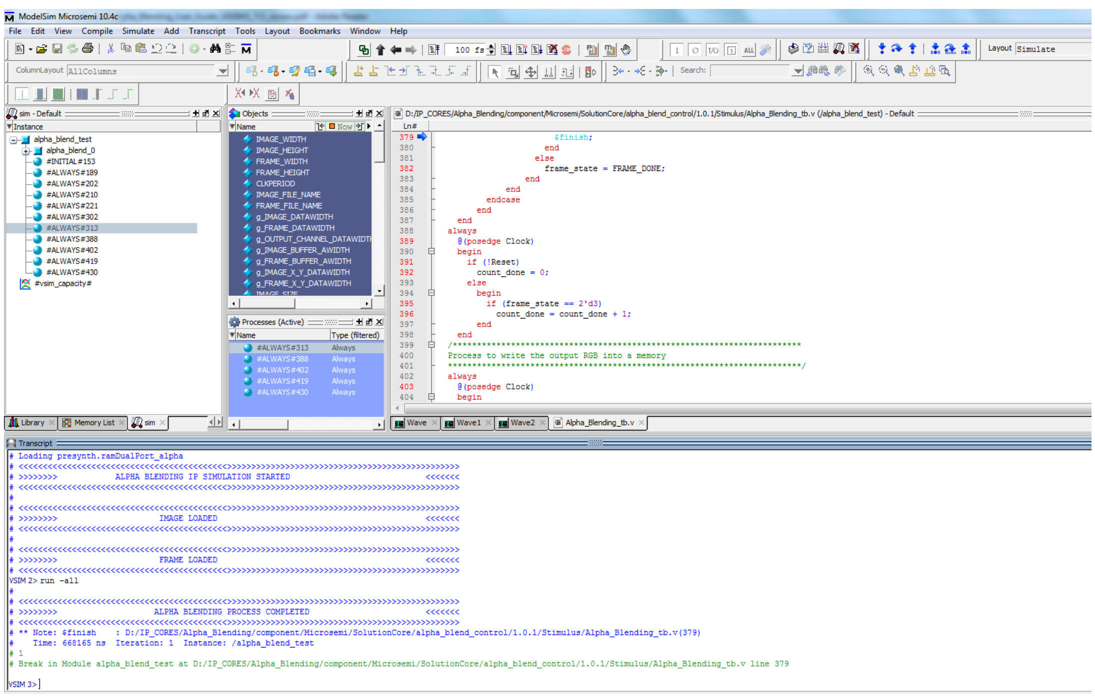
9. In the Stimulus hierarchy, expand **Work**, and right-click the **alpha_blend_test** file (Aplha_Blending_tb.v). Select **Simulate Pre-Synth Design**, then click **Open Interactively**.

Figure 16 • Open Interactively Option



The ModelSim tool appears with the test bench file loaded on to it as shown in the following figure.

Figure 17 • ModelSim Tool with Alpha Blending Testbench File



10. If the simulation is interrupted because of the runtime limit in the DO file, use the `run -all` command in the transcript window to complete the simulation.

After the simulation is completed, the testbench output image file appears in the simulation folder.

3.5 Resource Utilization

The alpha blending block is implemented on an M2S150T SmartFusion®2 System-on-Chip (SoC) FPGA in the FC1152 package) and PolarFire FPGA (MPF300TS_ES - 1FCG1152E package).

Table 5 • Resource Utilization Report

Resource	Usage
DFFs	1347
4-Input LUTs	1544
MACC	12
RAM1Kx18	7
RAM64x18	0

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

50200641