UG0641
User Guide
Alpha Blending

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200641. 6.0 7/21

Contents

1 Revision History 1
1.1 Revision 6.0 1
1.2 Revision 5.0 1
1.3 Revision 4.0 1
1.4 Revision 3.0 1
1.5 Revision 2.0 1
1.6 Revision 1.0 1
2 Alpha Blending 2
2.1 Key Features 2
2.2 Supported Families 2
2.3 Hardware Implementation 2
2.4 License 3
2.4.1 Encrypted 3
2.4.2 RTL 3
2.5 Inputs and Outputs 3
2.6 Configuration Parameters 5
2.7 Timing Diagrams 6
2.8 Resource Utilization 6

Figures

Figure 1 Block Diagram of Alpha Blending IP 2
Figure 2 Timing Diagram of Alpha Blending IP 6

Tables

Table 1 Inputs and Outputs of Alpha Blending IPfor Native Video Interface . . . 3
Table 2 Input and Output Ports for AXI4 Stream Video Interface . . . 4
Table 3 Input and Output Ports for AXI4-Lite Interface . . . 5
Table 4 Register Map and Description . . . 5
Table 5 Configuration Parameters . . . 5
Table 6 Resource Utilization of PolarFire . . . 6
Table 7 Resource Utilization of SmartFusion2 . . . 6

1 Revision History

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

1.1 Revision 6.0

The following is a summary of the changes in this revision.

- Added Key Features, page 2, Supported Families, page 2, and Key Features, page 2.
- Added License, page 3, Encrypted, page 3, and RTL, page 3.
- Updated Table 1, page 3 and Table 5, page 5.
- Added Table 2, page 4, Table 3, page 5, and Table 4, page 5.

1.2 Revision 5.0

All the sections were updated in revision 5.0 of this document.

1.3 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, page 6.

1.4 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.

1.5 Revision 2.0

In revision 2.0 of this document, the SAR 76066 was updated.

1.6 Revision 1.0

Revision 1.0 was the first publication of this document.

2 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.

For example, an 8-bit alpha input can represent 256 levels of transparency with a value of 0 denoting that the image is completely transparent and a value of 255 denoting that the image is completely opaque. Alpha blending defines the transparency of individual images when blended with the background image.

The following are the equations for pixel-wise alpha blending:

$$R_{out} = R_{in1} \times (1 - \alpha_{in}) + R_{in2} \times \alpha_{in}$$

$$G_{out} = G_{in1} \times (1 - \alpha_{in}) + G_{in2} \times \alpha_{in}$$

$$B_{out} = B_{in1} \times (1 - \alpha_{in}) + B_{in2} \times \alpha_{in}$$

Where,

R_{in1}, G_{in1}, and B_{in1} represent the red, blue, and green values of image1

R_{in2}, G_{in2}, and B_{in2} represent the red, blue, and green values of image2

R_{out}, G_{out}, and B_{out} represent the red, blue, and green values of the output image

α_{in} is the input alpha value

2.1 Key Features

- Supports per pixel mixture of two video streams
- Supports data width of 8,10, and 12
- Supports Native and AXI4 Stream Video Interface for video data transfer
- Supports Native and AXI4-Lite Configuration Interface for parameter modification

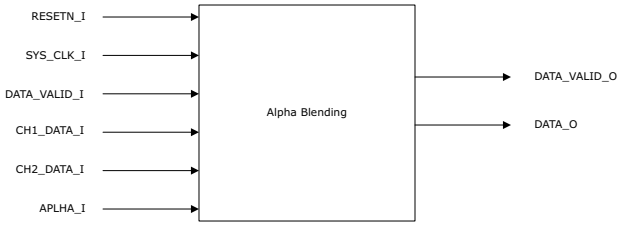
2.2 Supported Families

- PolarFire® SoC
- PolarFire®
- RTG4™
- IGLOO®2
- SmartFusion®2

2.3 Hardware Implementation

The following figure shows the hardware implementation of alpha blending.

Figure 1 • **Block Diagram of Alpha Blending IP**



When the DATA_VALID_I signal goes high, the R, G, and B values of the output is computed according to the preceding equation.

DATA_O is valid when DATA_VALID_O (which is equivalent to DATA_VALID_I with two clock cycle delay) goes high.

2.4 License

Alpha Blending clear RTL is license locked, and the encrypted RTL is available for free.

2.4.1 Encrypted

Complete RTL code is provided for the core, allowing it to be instantiated with the SmartDesign tool.

Simulation, synthesis, and layout can be performed within Libero® System-on-Chip (SoC). The RTL code is encrypted for the core.

2.4.2 RTL

Complete RTL source code is provided for the core.

2.5 Inputs and Outputs

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

Table 1 • **Input and Output for Native Video Interface**

Signal Name	Direction	Width	Description
RESETN_I	Input	1-bit	The active-low asynchronous reset signal to design
SYS_CLK_I	Input	1-bit	System clock
DATA_VALID_I	Input	1-bit	Input data valid
CH1_DATA_I	Input	3 × G_PIXEL_WIDTH	Channel1 input data CH1_DATA_I [3 × g_PIXEL_WIDTH-1] to CH1_DATA_I [2 × g_PIXEL_WIDTH] represents R value of input1 CH1_DATA_I [2 × g_PIXEL_WIDTH-1] to CH1_DATA_I [g_PIXEL_WIDTH] represents G value of input1 CH1_DATA_I [g_PIXEL_WIDTH-1] to CH1_DATA_I [0] represents B value of input1
CH2_DATA_I	Input	3 × G_PIXEL_WIDTH	Channel2 input data CH2_DATA_I [3 × g_PIXEL_WIDTH-1] to CH2_DATA_I [2 × g_PIXEL_WIDTH] represents R value of input2 CH2_DATA_I [2 × g_PIXEL_WIDTH-1] to CH2_DATA_I [g_PIXEL_WIDTH] represents G value of input2 CH2_DATA_I [g_PIXEL_WIDTH-1] to CH2_DATA_I [0] represents B value of input2
ALPHA_I	Input	G_PIXEL_WIDTH	Alpha input
DATA_VALID_O	Output	1-bit	Asserted when output data is valid

Table 1 • **Input and Output for Native Video Interface *(continued)***

Signal Name	Direction	Width	Description
DATA_O	Output	3 × G_PIXEL_WIDTH	Output data DATA_O [3 × g_PIXEL_WIDTH-1] to DATA_O [2 × g_PIXEL_WIDTH] represents R output DATA_O [2 × g_PIXEL_WIDTH-1] to DATA_O [g_PIXEL_WIDTH] represents G output DATA_O [g_PIXEL_WIDTH-1] to DATA_O [0] represents B output

Table 2 • **Input and Output Ports for AXI4 Stream Video Interface**

Port Name	Type	Width	Description
RESETN_I	Input	1 bit	Active low asynchronous reset signal to design
SYS_CLK_I	Input	1 bit	System clock
TREADY_O	Output	1 bit	Output target ready
TDATA_I	Input	3* G_PIXEL_WIDTH bit	Channel1 Input Video Data
TDATA_I_0	Input	3* G_PIXEL_WIDTH bit	Channel2 Input Video Data
TVALID_I	Input	1 bit	Channel1 Input Video Valid
TVALID_I_0	Input	1 bit	Channel2 Input Video Valid
TUSER_I	Input	4 bits	Bit 0 = frame end Bit 1 = unused Bit 2 = unused Bit 3 = unused
TDATA_O	Output	3 *G_PIXEL_WIDTH bit	Output Video Data
TVALID_O	Output	1 bit	Output Video Valid
TUSER_O	Output	4 bits	Bit 0 = frame end Bit 1 = unused Bit 2 = unused Bit 3 = unused
TLAST_O	Output	1 bit	Output Video End of Frame
TSTRB_O	Output	G_DATA_WIDTH/8	Output Video Data strobe
TKEEP_O	Output	G_DATA_WIDTH/8	Output Video Data Keep

Table 3 • Input and Output Ports for AXI4-Lite Interface

Port Name	Type	Width	Description
AXI_CLK_I	Input	1 bit	AXI4-Lite clock signal
AXI_RESETN_I	Input	1 bit	AXI4-Lite asynchronous reset signal
AXI_AWVALID_I	Input	1 bit	AXI4-Lite Write address Valid
AXI_AWREADY_O	Output	1 bit	AXI4-Lite Write address ready
AXI_AWADDR_I	Input	32 bits	AXI4-Lite Write address
AXI_WDATA_I	Input	32 bits	AXI4-Lite Write Data
AXI_WVALID_I	Input	1 bit	AXI4-Lite Write Data Valid
AXI_WREADY_O	Output	1 bit	AXI4-Lite Write Data Ready
AXI_BVALID_O	Output	1 bit	AXI4-Lite Write Response Valid
AXI_BREADY_I	Input	1 bit	AXI4-Lite Write Response Ready
AXI_BRESP_O	Output	2 bits	AXI4-Lite Write Response
AXI_ARVALID_I	Input	1 bit	AXI4-Lite Read Address Valid
AXI_ARREADY_O	Output	1 bit	AXI4-Lite Read Address Ready
AXI_ARADDR_I	Input	32 bits	AXI4-Lite Read Address
AXI_RDATA_O	Output	32 bits	AXI4-Lite Read Data
AXI_RVALID_O	Output	1 bit	AXI4-Lite Read Data Valid
AXI_RREADY_I	Input	1 bit	AXI4-Lite Read Data Ready
AXI_RRESP_O	Output	2 bits	AXI4-Lite Read Response

IP has one specific register through which users can dynamically control the operation of IP.

Table 4 • Register Map and Description

Address (hex)	Register name	Type	Description
0x014	ALPHA_ADDR	Read/Write	Alpha Input

2.6 Configuration Parameters

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

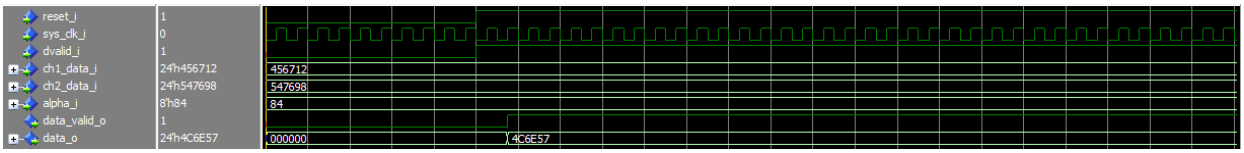
Table 5 • Configuration Parameters

Name	Description
Pixel Width	Bit width of each pixel
Video Interface	Native Video Interface and AXI4 Stream Video Interface
Configuration Interface	Native Configuration Interface and AXI4-Lite Configuration Interface

2.7 Timing Diagrams

The following figure shows the timing diagram of alpha blending IP.

Figure 2 • **Timing Diagram of Alpha Blending IP**



2.8 Resource Utilization

The alpha blending IP is implemented on SmartFusion®2 System-on-Chip (SoC) Field Programmable Gate Array (FPGA) device (M2S150T-1152 FC package) and PolarFire® FPGA (MPF300TS - 1FCG1152E package).

Table 6 • **Resource Utilization of PolarFire[1]**

Resource	Usage
DFFs	242
4-Input LUTs	273
MACC	6
RAM1Kx18	0
RAM64x18	0

1. When G_PIXEL_WIDTH = 8

Table 7 • **Resource Utilization of SmartFusion2[1]**

Resource	Usage
DFFs	242
4-Input LUTs	273
MACC	6
RAM1Kx18	0
RAM64x18	0

1. When G_PIXEL_WIDTH = 8