**BRAM IPs**

**BRAM Controller IPs**

1. axi\_interconnect\_0
   1. The main interconnect for all BRAM controllers. Enables an AXI interface with each BRAM controller so that memory can be communicated with via SDK.
2. axi\_bram\_ctrl\_0
   1. Connected with a standard AXI interface. Port A is connected to port A of blk\_mem\_gen\_0. Current address is: 0x00\_B000\_0000-0x00\_B007\_FFFF.
3. axi\_bram\_ctrl\_1
   1. Connected with a standard AXI interface. Port A is connected to port A of blk\_mem\_gen\_1. Current address is: 0x00\_B008\_0000-0x00\_B00F\_FFFF.
4. axi\_bram\_ctrl\_2
   1. Connected with a standard AXI interface. Port A is connected to port A of blk\_mem\_gen\_5. Current address is: 0x00\_B010\_0000-0x00\_B017\_FFFF.
5. axi\_bram\_ctrl\_3
   1. Connected with a standard AXI interface. Port A is connected to port A of blk\_mem\_gen\_2. Current address is: 0x00\_B018\_0000-0x00\_B018\_0FFF.
6. axi\_bram\_ctrl\_4
   1. Connected with a standard AXI interface. Port A is connected to port A of blk\_mem\_gen\_8. Current address is: 0x00\_B018\_1000-0x00\_B018\_1FFF.
7. axi\_bram\_ctrl\_5
   1. Connected with a standard AXI interface. Port A is connected to port A of blk\_mem\_gen\_9. Current address is: 0x00\_B018\_2000-0x00\_B018\_2FFF.

**BRAM IPs**

1. blk\_mem\_gen\_0

**I/O**

1. addrb [32-bit input]
   1. [IP: Interface\_0] out\_grad\_gamma\_addr\_ints\_ref\_0 → addrb
2. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
3. dinb [32-bit input]
   1. [IP: GradientsMulti\_1] din\_ints\_0 → dinb
4. doutb [32-bit output]
   1. doutb → ref\_img\_in\_0 [IP: MUX\_0]
5. enb [1-bit input]
   1. [IP: GradientsMulti\_1] ints\_ea\_0 → enb
6. weab [4-bit input]
   1. [IP: Interface\_0] out\_grad\_wea\_ints\_0 → weab

**Description**

BRAM 0 is connected to AXI controller 0 and has a size of 512k. This BRAM is used by the echo.c server on the ARM processor to write the pixel data of an image. It initially starts out as the reference image but alternates to a deformed image [image 2,4,6...]. The stored image data is then sent to the MUX IP, which acts as a ping-pong buffer, to send the correct data to either the GradientsMulti\_1 IP or the Gamma\_Imp\_0 IP.

1. blk\_mem\_gen\_1

**I/O**

1. addrb [32-bit input]
   1. [IP: Interface\_0] out\_gamma\_addr\_ints\_def\_0 → addrb
2. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
3. dinb [32-bit input]
   1. [IP: GradientsMulti\_1] din\_ints\_0 → dinb
4. doutb [32-bit output]
   1. doutb → def\_img\_in [IP: MUX\_0]
5. enb [1-bit input]
   1. [IP: GradientsMulti\_1] ints\_ea\_0 → enb
6. weab [4-bit input]
   1. [IP: Interface\_0] out\_grad\_wea\_ints\_0 → weab

**Description**

BRAM 1 is connected to AXI controller 1 and has a size of 512k. This BRAM is used by the echo.c server on the ARM processor to write the pixel data of an image. It initially starts out as the deformed image but alternates to a reference image [image 3,5,7...]. The stored image data is then sent to the MUX IP, which acts as a ping-pong buffer, to send the correct data to either the GradientsMulti\_1 IP or the Gamma\_Imp\_0 IP.

1. blk\_mem\_gen\_2

**I/O**

1. addrb [32-bit input]
   1. [IP: ParametersMulti\_0] addr\_0 → addrb
2. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
3. dinb [32-bit input]
   1. [IP: ParametersMulti\_0] param\_bram\_dout\_0 → dinb
4. doutb [32-bit output]
   1. doutb → data\_in\_0 [IP: ParametersMulti\_0]
5. enb [1-bit input]
   1. [IP: ParametersMulti\_0] ea\_0 → enb
6. web [4-bit input]
   1. [IP: ParametersMulti\_0] we\_0 → weab

**Description**

BRAM 2 is connected to AXI controller 3 and has a size of 4k. The echo.c server writes the user-defined parameters data to the BRAM that are needed to perform a specific type of DIC. It receives all parameters data and sends it to the ParametersMulti\_0 IP so that other IPs may have access to the data.

1. blk\_mem\_gen\_3 [Standalone]

**I/O**

1. addra [17-bits input]
   1. [IP: GradientsMulti\_1] addr\_grad\_x\_0 → addra
2. addrb [17-bits input]
   1. [IP: Gamma\_Imp\_0] addr\_grad\_x\_0 → addrb
3. clka [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
4. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
5. dina [32-bits input]
   1. [IP: GradientsMulti\_1] din\_grad\_x\_0 → dina
6. doutb [32-bits output]
   1. doutb → dout\_grad\_x\_0 [IP: Gamma\_Imp\_0]
7. ena [1-bit input]
   1. [IP: GradientsMulti\_1] grad\_ea\_0 → ena
8. enb [1-bit input]
   1. [IP: GradientsMulti\_1] grad\_ea\_0 → enb
9. wea [1-bit input]
   1. [IP: GradientsMulti\_1] grad\_wea\_0 → wea
10. web [1-bit input]
    1. [IP: GradientsMulti\_1] grad\_wea\_0 → web

**Description**

BRAM 3 is a standalone BRAM that has a size of 3,325,952 bits [or 415.744 KB]. It is used to save the X-coordinate gradient values for each pixel of the image that is computed by the GradientsMulti\_0 IP. Once the X-coordinate gradient values are computed by the IP, they are sent to BRAM 3 for storage and then sent to the Gamma\_Imp\_0 IP for further DIC.

1. blk\_mem\_gen\_4 [Standalone]

**I/O**

1. addra [17-bits input]
   1. [IP: GradientsMulti\_1] addr\_grad\_y\_0 → addra
2. addrb [17-bits input]
   1. [IP: Gamma\_Imp\_0] addr\_grad\_y\_0 → addrb
3. clka [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
4. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
5. dina [32-bits input]
   1. [IP: GradientsMulti\_1] din\_grad\_y\_0 → dina
6. doutb [32-bits output]
   1. doutb → dout\_grad\_y\_0 [IP: Gamma\_Imp\_0]
7. ena [1-bit input]
   1. [IP: GradientsMulti\_1] grad\_ea\_0 → ena
8. enb [1-bit input]
   1. [IP: GradientsMulti\_1] grad\_ea\_0 → enb
9. wea [1-bit input]
   1. [IP: GradientsMulti\_1] grad\_wea\_0 → wea
10. web [1-bit input]
    1. [IP: GradientsMulti\_1] grad\_wea\_0 → web

**Description**

BRAM 4 is a standalone BRAM that has a size of 3,325,952 bits [or 415.744 KB]. It is used to save the Y-coordinate gradients values for each pixel of the image that is computed by the GradientsMulti\_0 IP. Once the Y-coordinate gradients values are computed by the IP, they are sent to BRAM 4 for storage and then sent to the Gamma\_Imp\_0 IP for further DIC.

1. blk\_mem\_gen\_5

**I/O**

1. addrb [32-bit input]
   1. [IP: Gam\_Interface\_0] param\_addr → addrb
2. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
3. dinb [32-bit input]
   1. [IP: Results\_0] din\_0 → dinb
4. enb [1-bit input]
   1. [IP: Results\_0] ea\_0 → enb
5. web [4-bit input]
   1. [IP: Results\_0] we\_0 → web

**Description**

BRAM 5 is connected to AXI controller 2 and has a size of 512k. BRAM 5 is connected to the Results\_0 IP and it stores all of the results that were computed by the Gamma\_Imp\_0 IP such as Displacement X, Y, and Rotation Z. Once the values are saved in memory, the echo.c server performs multiple reads to extract the data and send it to the Python script on the PC.

1. blk\_mem\_gen\_6 [Standalone]

**I/O**

1. addra [16-bits input]
   1. [IP: SubsetCoordsMulti\_0] addr\_x\_0 → addra
2. addrb [16-bits input]
   1. [IP: Gamma\_Imp\_0] sub\_coord\_addr\_x\_0 → addrb
3. clka [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
4. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
5. dina [32-bits input]
   1. [IP: SubsetCoordsMulti\_0] din\_x\_0 → dina
6. doutb [32-bits output]
   1. doutb → x\_0[IP: Gamma\_Imp\_0]
7. ena [1-bit input]
   1. [IP: SubsetCoordsMulti\_0] ea\_x\_0 → ena
8. enb [1-bit input]
   1. [IP: Gamma\_Imp\_0] sub\_coord\_ea\_x\_0 → enb
9. wea [1-bit input]
   1. [IP: SubsetCoordsMulti\_0] wea\_x\_0 → wea
10. web [1-bit input]
    1. [IP: Gamma\_Imp\_0] sub\_coord\_we\_x\_0 → web

**Description**

BRAM 6 is a standalone BRAM that has a size of 1,545,600 bits [or 193.2 KB]. It is used to save the X subset coordinate values for each pixel of the subset within an image that is computed by the SubsetCoordsMulti\_0 IP. Once the X subset coordinate values are computed by the IP, they are sent to BRAM 6 for storage and then sent to the Gamma\_Imp\_0 IP for further DIC.

Subset X

1. blk\_mem\_gen\_7 [Standalone]

**I/O**

1. addra [16-bits input]
   1. [IP: SubsetCoordsMulti\_0] addr\_y\_0 → addra
2. addrb [16-bits input]
   1. [IP: Gamma\_Imp\_0] sub\_coord\_addr\_y\_0 → addrb
3. clka [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
4. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
5. dina [32-bits input]
   1. [IP: SubsetCoordsMulti\_0] din\_y\_0 → dina
6. doutb [32-bits output]
   1. doutb → y\_0[IP: Gamma\_Imp\_0]
7. ena [1-bit input]
   1. [IP: SubsetCoordsMulti\_0] ea\_y\_0 → ena
8. enb [1-bit input]
   1. [IP: Gamma\_Imp\_0] sub\_coord\_ea\_y\_0 → enb
9. wea [1-bit input]
   1. [IP: SubsetCoordsMulti\_0] wea\_y\_0 → wea
10. web [1-bit input]
    1. [IP: Gamma\_Imp\_0] sub\_coord\_we\_y\_0 → web

**Description**

BRAM 7 is a standalone BRAM that has a size of 1,545,600 bits [or 193.2 KB]. It is used to save the Y subset coordinate values for each pixel of the subset within an image that is computed by the SubsetCoordsMulti\_0 IP. Once the Y subset coordinate values are computed by the IP, they are sent to BRAM 7 for storage and then sent to the Gamma\_Imp\_0 IP for further DIC.

1. blk\_mem\_gen\_8

**I/O**

1. addrb [32-bit input]
   1. [IP: Coords\_Interace\_0] param\_addr\_0 → addrb
2. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
3. doutb [32-bit output]
   1. doutb → param\_dout\_0 [IP: Coords\_Interface\_0]
4. enb [1-bit input]
5. [IP: Coords\_Interface\_0] param\_ea\_0 → enb
6. web [4-bit input]
   1. [IP: Coords\_Interface\_0] param\_wea\_0 → web

**Description**

BRAM 8 is connected to AXI controller 4 and has a size of 4k. It is used similarly to BRAM 2. The echo.c server writes the user-defined parameters data to the BRAM that are needed to perform a specific type of DIC. It receives all parameters data and sends it to the Coords\_Interface\_0 IP so that other IPs may have access to the data. The Coords\_Interface\_0 IP needs some of the same parameters as defined in BRAM 2 but requires its own memory for reading and write control.

1. blk\_mem\_gen\_9

**I/O**

1. addrb [32-bit input]
   1. [IP: Gam\_Interface\_0] param\_addr\_0 → addrb
2. clkb [1-bit input]
   1. Hooked to the PL clock from zynq\_ultra\_ps\_e\_0 at 150 MHz.
3. doutb [32-bit output]
   1. doutb → param\_dout\_0 [IP: Gam\_Interface\_0]
4. enb [1-bit input]
   1. [IP: Gam\_Interface\_0] param\_ea\_0 → enb
5. web [4-bit input]
   1. [IP: Gam\_Interface\_0] param\_wea\_0 → web

**Description**

BRAM 9 is connected to AXI controller 5 and has a size of 4k. It is used similarly to BRAM 2. The echo.c server writes the user-defined parameters data to the BRAM that are needed to perform a specific type of DIC. It receives all parameters data and sends it to the Gam\_Interface\_0 IP so that other IPs may have access to the data. The Gam\_Interface\_0 IP needs some of the same parameters as defined in BRAM 2 but requires its own memory for reading and write control.