**Results IP**

**Inputs**

1. S00\_AXI
   1. [IP: axi\_interconnect\_1] M00\_AXI → S00\_AXI
2. gamma\_done\_0 [1-bit]
   1. [IP: Gamma\_Imp\_0] gam\_done\_0 → gamma\_done\_0
3. results\_done\_0 [1-bit]
   1. [IP: Gamma\_Imp\_0] results\_done\_0 → results\_done\_0
4. dis\_X\_0 [32-bits]
   1. [IP: Gamma\_Imp\_0] disp\_x\_0 → dis\_X\_0
5. dis\_Y\_0 [32-bits]
   1. [IP: Gamma\_Imp\_0] disp\_y\_0 → dis\_Y\_0
6. dis\_Z\_0 [32-bits]
   1. [IP: Gamma\_Imp\_0] disp\_Z\_0 → dis\_Z\_0
7. s00\_axi\_aclk [1-bit]
   1. [IP: zynq\_ultra\_ps\_e\_0] pl\_clk0 → s00\_axi\_aclk
8. s00\_axi\_aresetn [1-bit]
   1. [IP: rst\_ps8\_0\_100M] peripheral\_aresetn → s00\_axi\_aresetn

**Associated IPs (inputs):**

1. zynq\_ultra\_ps\_e\_0
2. rst\_ps8\_0\_100M
3. axi\_interconnect\_1
4. Gamma\_Imp\_0

**Outputs**

1. addr\_0 [32-bits]
   1. addr\_0 → addrb [IP: blk\_mem\_gen\_5]
2. we\_0 [4-bits]
   1. we\_0 → web [IP: blk\_mem\_gen\_5]
3. ea\_0 [1-bit]
   1. ea\_0 → enb [IP: blk\_mem\_gen\_5]
4. results\_done\_0 [32-bits]
   1. results\_done\_0 → probe\_in# [IP: VIO]
5. din\_0 [32-bits]
   1. din\_0 → dinb [IP: blk\_mem\_gen\_5]
   2. din\_0 → probe\_in# [IP: VIO]

**Associated IPs (outputs):**

1. blk\_mem\_gen\_5 [BRAM 5]
2. VIO

**IP Description**

The Results\_0 IP is very simple and it essentially acts as a buffer and handles the addressing of the results to memory [BRAM 5]. The IP consists of six inputs: clock, gamma\_done, results\_done, dis\_X, dis\_Y, and dis\_Z. Its six outputs are: addr, we, ea, result\_done, din, and save\_done. The results IP is guided by the Gamma\_Imp\_0 IP that is responsible for computing all of the correlation results: displacement X, displacement Y, and Rotation Z (referred to as displacement z in the IPs). When two frames are finished with their correlation, the Gamma\_Imp\_0 IP sets the wire “results\_done” to a 1 which tells the results IP to start saving the data it is sent. The results IP then jumps to state one where it writes the displacement X values, then enters two “dummy” states so that the IP is given enough time to write the data to BRAM. \*NOTE: these dummy states can be reduced; it takes one clock cycle for a write operation so the dummy states can be reduced from two to one, potentially even to zero. They were added to aid in debugging and as a safety net. This same trend continues for the Y and Z values. In the second to last state, the IP then jumps back to state 0 where it will wait for the next round of results from the Gamma IP. If the Gamma IP sends the signal “gamma\_done” as a 0, then the results IP will wait in state 0 as it means the IP isn’t finished computing all of the frames yet. If the “gamma\_done” signal is set to a 1, that means the Gamma\_Imp\_0 IP has finished processing all of the DIC on all available frames and all of the results are ready. When this happens, the results IP jump to the last state where it signals that it is complete. The “Save\_Done” signal is defined in the IP as an output, however, it is not visible in the Vivado block diagram because it acts as an internal AXI slave register that writes its value to where the echo.c server can read it directly from the IP. Once that signal is a ‘1’, the echo.c server will read all of the results data and send it to the Python script that is hosted on the PC.