**Coords\_Interface IP**

**Inputs**

1. S00\_AXI
   1. [IP: axi\_interconnect\_1] M07\_AXI → S00\_AXI
2. coord\_done\_0 [1-bit]
   1. [IP: SubsetCoordsMulti\_0] sub\_done\_0 → coord\_done\_0
3. parameters\_done\_0 [1-bit]
   1. [IP: ParametersMulti\_0] param\_done\_0 → parameters\_done\_0
4. param\_dout\_0 [32-bits]
   1. [IP: blk\_mem\_gen\_8 ] doutb → param\_dout\_0
5. coord\_subset\_number\_0 [32-bits]
   1. [IP: SubsetCoordsMulti\_0] subset\_counter\_0 → coord\_subset\_number\_0
6. coord\_new\_susbet\_0 [1-bit]
   1. [IP: SubsetCoordsMulti\_0] coord\_new\_susbet\_0 → coord\_new\_susbet\_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. SubsetCoordsMulti\_0
5. ParametersMulti\_0
6. blk\_mem\_gen\_8 [BRAM 8]

**Outputs**

1. param\_ea\_0 [1-bit]
   1. param\_ea\_0 → enb [IP: blk\_mem\_gen\_8]
2. param\_wea\_0 [4-bits]
   1. param\_wea\_0 → web [IP: blk\_mem\_gen\_8]
3. param\_addr\_0 [32-bits]
   1. param\_addr\_0 → addrb [IP: blk\_mem\_gen\_8]
   2. param\_addr\_0 → probe\_in# [IP: VIO]
4. coord\_cx\_0 [32-bits]
   1. coord\_cx\_0 → subset\_centerpoint\_x\_0 [IP: SubsetCoordsMulti\_0]
   2. coord\_cx\_0 → probe\_in# [IP: VIO]
5. coord\_cy\_0 [32-bits]
   1. coord\_cy\_0 → subset\_centerpoint\_y\_0 [IP: SubsetCoordsMulti\_0]
   2. coord\_cy\_0 → probe\_in# [IP: VIO]
6. subset\_size\_0 [32-bits]
   1. subset\_size\_0 → subset\_size\_0 [IP: SubsetCoordsMulti\_0]
   2. subset\_size\_0 → probe\_in# [IP: VIO]
7. half\_subset\_size\_0 [32-bits]
   1. half\_subset\_size\_0 → half\_subset\_size\_0 [IP: SubsetCoordsMulti\_0]
   2. half\_subset\_size\_0 → probe\_in# [IP: VIO]
8. subset\_shape\_0 [32-bits]
   1. subset\_shape\_0 → subset\_shape\_0 [IP: SubsetCoordsMulti\_0]
   2. subset\_shape\_0 → probe\_in# [IP: VIO]
9. coord\_interface\_done\_0 [1-bit]
   1. coord\_interface\_done\_0 → coord\_interface\_done\_0 [IP: SubsetCoordsMulti\_0]
   2. coord\_interface\_done\_0 → probe\_in# [IP: VIO]

**Associated IPs (outputs):**

1. blk\_mem\_gen\_8 [BRAM 8]
2. SubsetCoordsMulti\_0
3. VIO

**IP Description**

The Coords\_Inteface\_0 IP is responsible for sending a few dedicated parameters to the SubsetCoordsMulti\_0 IP. The parameters the Subset Coordinates IP requires are the X and Y subset center point coordinates, subset size, half subset size, and subset shape. This IP, along with Gam\_Interface\_0, were created because we cannot have multiple IP’s driving addresses to a single BRAM module that contains the user-defined parameters data. The solution around this was to create a few “interface” IPs that handle this addressing between their “parent” IPs. The user-defined parameters file represents a very small and insignificant amount of data, so we saw no problem with having this data duplicated into 3 separate BRAM modules. BRAM 8 is responsible for the parameter data that the Coords\_Interface\_0 IP uses. The IP waits for the parameter data to be finished with its writing of data into the BRAMs memory, after which it goes through a sequence of states that pull out the information required (listed above). The Coords\_Inteface IP and the SubsetCoordsMulti IP both start at the same time. SubsetCoordsMulti requests a new subset to process which signals the Coords\_Interface IP to start sending data over. It starts with reading the X center point and is followed by two “dummy” states so the information is correctly set to the signals because a read operation from BRAM takes two clock cycles. \*NOTE: If URAM is being used instead of BRAM, the number of dummy states can be reduced from two to one because URAM requires one clock cycle for each read operation. The numbers that handle the addressing are dependent on the way the “Subsets.txt” file is defined (listed below). Once all information is ready in the Coords\_Interface IP it will signal that it's finished processing a subset. After the required data for the first subset is sent over to the SubsetCoordsMulti IP, the same IP will process all of the pixels contained within the subset based on the information it was given. The Coords\_Interface IP will jump back to state zero and wait for the SubsetCoordsMulti IP to signal that it is ready for another set of subset information. This will continue for as many subsets exist within the defined Subsets.txt file.

**Example Subsets.txt file**

1) 232 [Image height]

2) 448 [Image width]

3) 103936 [Number of pixels per image]

4) 3325952 [Number of bits per image]

5) 4 [Number of subsets]

6) 0 [Optimization method]

7) 0 [Correlation method]

8) 1 [Subset shape (1 denotes a square shape)]

9) 30 [Subset centerpoint (x-coordinate)]

10) 30 [Subset centerpoint (y-coordinate)]

11) 3 [Subset size]

12) 1 [Half subset size (floor function of the subset size)]

13) 0 [Subset shape (0 denotes a circle shape)]

14) 10 [Subset centerpoint (x-coordinate)]

15) 10 [Subset centerpoint (y-coordinate)]

16) 3 [Subset radius]

17) 9 [Subset radius squared]

18) 0 [Subset shape (0 denotes a circle shape)]

19) 120 [Subset centerpoint (x-coordinate)]

20) 120 [Subset centerpoint (y-coordinate)]

21) 2 [Subset radius]

22) 4 [Subset radius squared]

23) 1 [Subset shape (1 denotes a square shape)]

24) 100 [Subset centerpoint (x-coordinate)]

25) 80 [Subset centerpoint (y-coordinate)]

26) 5 [Subset size]

27) 2 [Half subset size (floor function of the subset size)]