1)

1. We can have an extra column in each inner ROB entries to denote if the data should be forwarded.
2. If Instruction I2 is true dependent on I1 and I1 is already dispatched, I1 will be renamed to the corresponding ROB entry in F-RAT table.
3. While processing I2, the decode stage will know I2 dependency over I1 from the F-RAT table and it can set the forward column of I1 ROB table entry true.
4. So when the inner ROB receives the data, based on its forward column, it can forward data to required cluster.

2)

1. Whenever an instruction is dispatched to local ROB entry, the pointer to this entry can be mapped in Shared ROB with the offset value.
2. The instruction in local ROB are retired to respective PRF while shared ROB retires the data to R-RAT.
3. When an instruction is retired in local ROB, corresponding data is committed in PRF.
4. When the same instruction is retired in shared ROB, the pointer to the PRF entry of the cluster is committed in R-RAT.

3)

1) Since each cluster knows if its output needed to be forwarded or not, it will be efficient if all the clusters are connected in a single bus.
2) Using more than one common bus does not increases much performance but it uses more power than using first one.
3) Each cluster can be connected together with a single bus.

4)

In addition to Normal RAT Table, there must be addition column for the table to point the specific cluster . So an entry in RAT table will consist of mapping architectural register to specific cluster and to the specific PRF entry of the cluster. For Eg.

RAT[R1] = CLUSTER1->P0

5)

It is better to make PRF access issue-bound as it will be easier to fetch data than dispatch-bound. If it is dispatch-bound, each cluster will need extra component to access data from other PRF. If its issue-bound, a central mechanism can be implemented to fetch data from PRF of each cluster.

6)

1) If there is any exception in any cluster, while retiring to PRF, the ROB of specific cluster will flush its queue till the exception caused instruction.
2) Then the specific ROB will notify the shared ROB about its exception status and the offset of the instruction.

3) Then, the shared ROB will flush its queue until the offset.
4) the shared ROB will notify all other clusters to flush the instructions based on its ROB entry

7)

1) BIS should be implemented as Global unit because if it is implemented in branch unit, allocating branch tag for other cluster instruction is not efficient one.
2) For every instruction, a branch tag will be associated in shared ROB.
3) Whenever mis-speculation happens, the associated branch tag can fetched from global BIS which points to the instruction address in shared ROB.
4) The shared ROB flushes the later queue and broadcasts to all other cluster ROB to flush its queue.

8)

1) Communication is needed between clusters during forwarding of data.
2) Since this architecture is out-of-order execution, The cluster does not know about execution of other clusters.
3) It needs to communicate with the other cluster to store or load the required operand
4) One cycle delay is incurred during forwarding results.
5) One of the benefits of this architecture is that each cluster does not need to forward data every time after it is executed which wastes power. It will be forwarded only when it is required which saves power.