**This is the design of the DPDK hash table:**

few parameters

TOT\_NUM\_ENTREES

NUM\_BUCKET\_ENTREES (value used is 8)

Creating the hash table

1) table = allocate memory for TOT\_NUM\_ENTREES / NUM\_BUCKETS

adding key (key,signature(hashvalue),value)

1) primary\_idx = signature/NUM\_BUCKETS -O(1)

seconday\_idx = alt\_hash\_value(signature)/NUM\_BUCKETS

2) check if key is already present in the primary index -O(1)

loop NUM\_BUCKET\_ENTREES times in table[primary\_idx]

check if key is present ... if yes then return that index

3) check if key is already present in the secondary index -O(1)

loop NUM\_BUCKET\_ENTREES times in table[seconday\_idx]

check if key is present ... if yes then return that index

4) check if primary bucket is full.If not -O(1) as ring buffer keeps count

add key into the primary bucket ring (queue insert)

5) if the primary bucket is full. move one of the entries to its secondary bucket (This operation could cascade ) // This is may not be O(1) as multiple cascading evictions could take place.

**Amos:** Explanation seems wrong in the Point 5, please check or if the sentence is not please go ahead and fix it?

fetching key

1) primary\_idx = signature/NUM\_BUCKETS -O(1)

seconday\_idx = alt\_hash\_value(signature)/NUM\_BUCKETS

2) check if key is already present in the primary index -O(1)

loop NUM\_BUCKET\_ENTREES times in table[primary\_idx]

check if key is present ... if yes then return that index

3) check if key is already present in the secondary index -O(1)

loop NUM\_BUCKET\_ENTREES times in table[seconday\_idx]

check if key is present ... if yes then return that index

deleting from the queue

1) primary\_idx = signature/NUM\_BUCKETS -O(1)

seconday\_idx = alt\_hash\_value(signature)/NUM\_BUCKETS

2) check if key is already present in the primary index

loop NUM\_BUCKET\_ENTREES times in table[primary\_idx]

check if key is present ... if yes set that index as free slot and enqueue it back to the ring buffer -O(1)

3) check if key is already present in the secondary index

loop NUM\_BUCKET\_ENTREES times in table[seconday\_idx]

check if key is present ... if yes set that index as free slot and enqueue it back to the ring buffer -O(1)