Algorithm

- `SPN.recover_round_keys` employes finding the optimal masks which result in the key blocks as specified in the argument `non_zero`
- For each output sbox block position `2**sbox_size` keys are tested to get putative output for the previous round
- the parity of the calculated optimal mask bits is checked to get its bias and for the putative key bits for which it matches the calculated optimal bias, is taken as the bits in the key at that position (note that we can take the absolute maximum bias since we were calculated linear combinations with maximal absolute bias)
- The above algo is extended to get all the round keys afterwards (additional care of pbox is taken afterwards)
- The function `SPN.dec` decrypt upto the given rounds as many round_keys are provided. Rest all functions are the functions we would see in a regular SPN
- It requires more than $1/\beta^2$ plaintext,ciphertext pairs to get the correct combination with given bias where β is the calculated bias