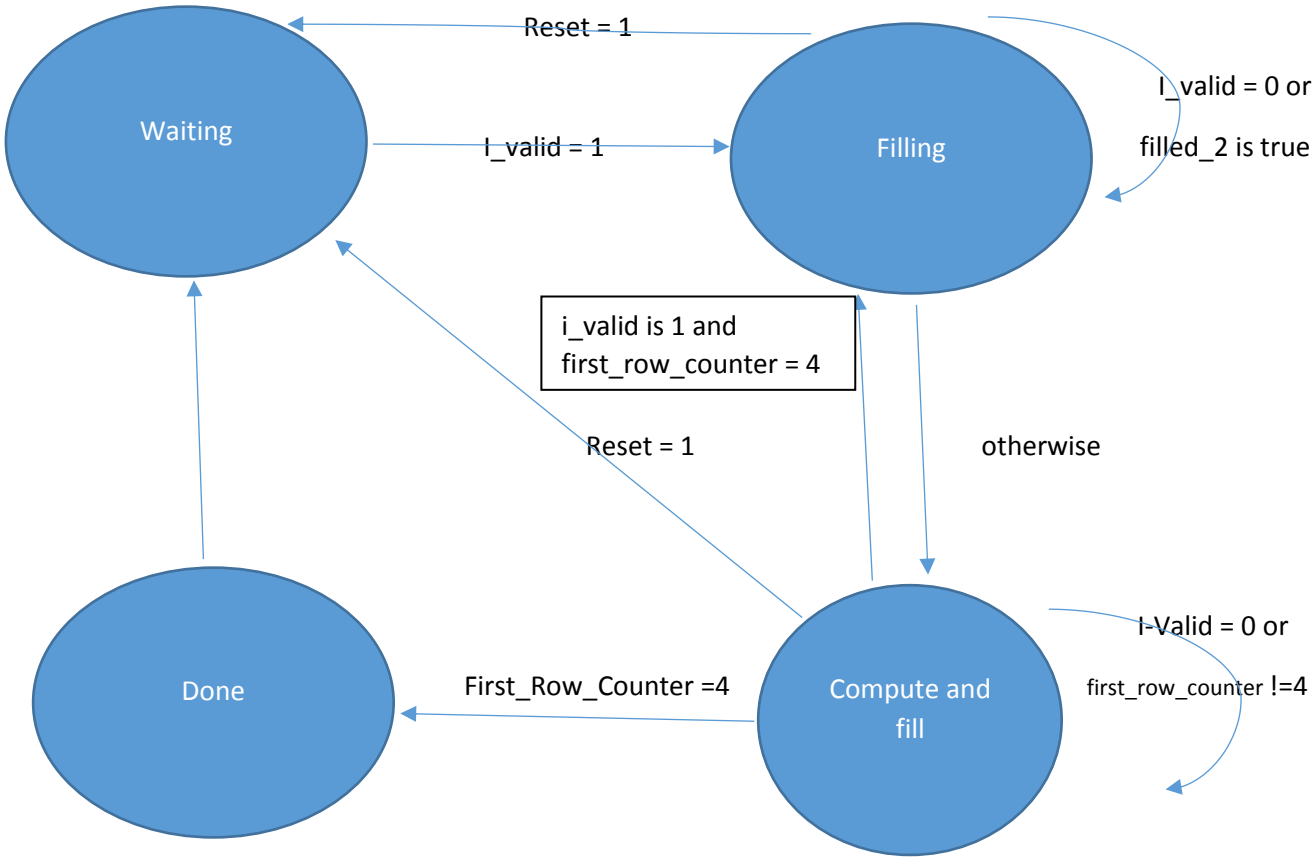
Mflaifel, bnassirz



Waiting : The reset state. We set everything to the initial value. It stays in this state until i_valid is one. If i_valid is 1 we capture i_data and go to the next state filling.

Filling: This is the state that the first 2 rows of the matrix gets filled for the first time. We keep filling the matrix in this state until we have filled up to the end of the 2nd row, then go to compute and fill state.

Compute and fill: compute and fill state both increments the counter for every cell in the matrix, now that we have enough previous rows to start the computation. It also stores new inputs in the matrix when i_valid is high. It keeps track of how many times we "re-used" the first row, so that It can know when the matrix is done using a signal "first_row_counter". When we have done the computation for every entry in the matrix, we can go to done state and output the count. However, if we are done but we also detect i_valid is high, that means there is a new input for the next matrix already, so o_done will be high and counter will hold the value from the previous matrix for 1 clock cycle only, because both will be reset as soon as we go into fill (next clock cycle).

Done state: outputs the counter and keeps o_done high. We only stay in done for 1 clock cycle then we go to wait (but wait doesn't reset o_done until it sees i_valid is high) that way we are meeting both

requirements of keeping o_done high until we get a new i_valid and we also process If i_valid is high right after we are done computing (this is achieved by the transition from compute and fill back to fill).