mac_row

```
module mac_row (clk, reset, out_s, in_w, in_n, valid, inst_w);

  parameter bw = 4;
  parameter psum_bw = 16;
  parameter col = 8;

  input  clk, reset;
  output [psum_bw*col-1:0] out_s;
  input  [bw-1:0] in_w;
  input  [psum_bw*col-1:0] in_n;
  output [col-1:0] valid;
  input  [1:0] inst_w;


  reg [bw-1:0] in_w_delayed;

  always @(posedge clk) begin
    if (reset)
        in_w_delayed <= 0;
    else
        in_w_delayed <= in_w;   // Delay input by 1 cycle
  end
    wire [(col+1)*bw-1:0] temp;
  wire [(col+1)*2-1:0] temp_inst;

  assign temp[bw-1:0] = in_w_delayed;

   assign temp_inst[1:0] = inst_w;

  genvar i;
  generate
    for (i=1; i < col+1 ; i=i+1) begin : col_num
        mac_tile #(.bw(bw), .psum_bw(psum_bw)) mac_tile_instance (
            .clk(clk),
            .reset(reset),
            .in_w( temp[bw*i-1 : bw*(i-1)] ),
            .out_e( temp[bw*(i+1)-1 : bw*i] ),
            .inst_w( temp_inst[2*i-1 : 2*(i-1)] ),
            .inst_e( temp_inst[2*(i+1)-1 : 2*i] ),
            .in_n( in_n[psum_bw*i-1 : psum_bw*(i-1)] ),
            .out_s( out_s[psum_bw*i-1 : psum_bw*(i-1)] )
        );
```

```
        assign valid[i-1] = temp_inst[2*i+1];
    end
  endgenerate

endmodule
```