We are generating a 64-to-1 multiplexer hierarchically in Verilog. We have generated 2-to-1 multiplexer, 4-to-1 multiplexer, 8-to-1 multiplexer defined as follows:

module mux2_1(
input in1,
input in2,
input select,
output out
);

```
  assign out = select ? in2 : in1;
```

endmodule

module mux4_1(
input [3:0] in,
input [1:0] select,
output out
);

```
  wire out1, out2;

  mux2_1 mux1 (
      .in1(in[0]),
      .in2(in[1]),
      .select(select[0]),
      .out(out1)
  );

  mux2_1 mux2 (
      .in1(in[2]),
      .in2(in[3]),
      .select(select[0]),
      .out(out2)
  );

  mux2_1 mux3 (
      .in1(out1),
      .in2(out2),
      .select(select[1]),
```

```
        .out(out)
    );
```

endmodule

module mux8_1(
input [7:0] in,
input [2:0] select,
output out
);

```
  wire out1, out2;

  mux4_1 mux1 (
      .in(in[3:0]),
      .select(select[1:0]),
      .out(out1)
  );

  mux4_1 mux2 (
      .in(in[7:4]),
      .select(select[1:0]),
      .out(out2)
  );

  mux2_1 mux3 (
      .in1(out1),
      .in2(out2),
      .select(select[2]),
      .out(out)
  );
```

endmodule

Please include the previous module(s) in your response and use them to hierarchically generate a 16-to-1 multiplexer defined as:
module mux16_1(in[15:0], select, out);
//Insert code here
endmodule

```
  module mux2_1(
      input  in1,
      input  in2,
      input  select,
      output out
  );

      assign out = select ? in2 : in1;

  endmodule

  module mux4_1(
```

```verilog
    input [3:0] in,
    input [1:0] select,
    output out
);

    wire out1, out2;

    mux2_1 mux1 (
        .in1(in[0]),
        .in2(in[1]),
        .select(select[0]),
        .out(out1)
    );

    mux2_1 mux2 (
        .in1(in[2]),
        .in2(in[3]),
        .select(select[0]),
        .out(out2)
    );

    mux2_1 mux3 (
        .in1(out1),
        .in2(out2),
        .select(select[1]),
        .out(out)
    );

endmodule

module mux8_1(
    input [7:0] in,
    input [2:0] select,
    output out
);

    wire out1, out2;

    mux4_1 mux1 (
        .in(in[3:0]),
        .select(select[1:0]),
        .out(out1)
    );

    mux4_1 mux2 (
        .in(in[7:4]),
        .select(select[1:0]),
        .out(out2)
    );

    mux2_1 mux3 (
        .in1(out1),
        .in2(out2),
        .select(select[2]),
        .out(out)
    );
```

```verilog
endmodule

module mux16_1(
    input [15:0] in,
    input [3:0] select,
    output out
);

    wire out1, out2;

    mux8_1 mux1 (
        .in(in[7:0]),
        .select(select[2:0]),
        .out(out1)
    );

    mux8_1 mux2 (
        .in(in[15:8]),
        .select(select[2:0]),
        .out(out2)
    );

    mux2_1 mux3 (
        .in1(out1),
        .in2(out2),
        .select(select[3]),
        .out(out)
    );

endmodule
```